CRT-2001-28

Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

META-HEURISTICS FOR A CLASS OF DEMAND-RESPONSIVE TRANSIT SYSTEMS

by

Teodor Gabriel Crainic
Federico Malucelli
Maddalena Nonato
François Guertin

August 2001

102-02791/

PUBLICATION
CRT-2001-28

META-HEURISTICS FOR A CLASS OF DEMAND-RESPONSIVE TRANSIT SYSTEMS*

by

Teodor Gabriel Crainic[1]
Federico Malucelli[2]
Maddalena Nonato[3]
François Guertin[4]

* This research was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC), the Fonds pour la Formation de Chercheurs et l'Aide à la Recherche (FCAR), the Quebec-Italy cooperation program of the Ministry for Foreign Affairs of Italy, the Ministry for International Relations of the Province of Quebec and the *Progetto strategico: Metodologie per la logistica* of the CNR Italia.

[1] Département de management et technologie, Université du Québec à Montréal, et Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7 (theo@crt.umontreal.ca)

[2] DEI, Politecnico di Milano, Piazza L. da Vinci 32, 20133 Milano, Italy (malucell@elet.polimi.it)

[3] DIEI, Università di Perugia, Piazza Univesità n.1, 06100 Perugia, Italy (nonato@istel.ing.unipg.it)

[4] Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3T 3J7 (guertin@crt.umontreal.ca)

Dépôt légal – Bibliothèque nationale du Québec
Bibliothèque nationale du Canada, 2001

ABSTRACT

The demand adaptive systems studied in this paper attempt to offer demand-responsive services within the framework of traditional scheduled bus transportation: users call to request service between two given points and, in so doing, induce detours in the vehicle routes; at the same time, thought, a given set of compulsory stops is always served according to a predefined schedule regardless of the current set of active requests. The model developed to select requests and determine the routing of the vehicle yields a difficult formulation but with a special structure that may be used to develop efficient algorithms. In this paper, we develop, test , and compare several solution strategies that belong to two general metaheuristic classes, memory-enhanced greedy randomized multi-start constructive procedures and tabu search methods. Hybrid meta-heuristics combining the two methods are also analyzed.

Key words: meta-heuristics, greedy randomized multi-start constructive procedures, tabu search, hybrids, public transport, demand adaptive systems.

RÉSUMÉ

L'objectif des systèmes adaptatifs de transport en commun est d'offrir des services qui s'ajustent selon les demandes reçues mais fonctionnant dans le cadre traditionnel des lignes et des horaires. Dans ces cas, les utilisateurs font des demandes pour des déplacements entre deux points et induisent ainsi des détours dans les routes des véhicules. Cependant, les véhicules doivent toujours passer à certaines stations à des moments précisés dans les horaires. Le modèle mis au point pour effectuer la sélection des requêtes et la construction des routes donne naissance à une formulation difficile mais présentant une structure particulière qui permet le développement d'algorithmes efficaces. Nous présentons et analysons plusieurs méthodes appartenant soit à la classe des procédures constructives « greedy randomized multi-start », soit à celle des recherches avec tabous. Des méta-heuristiques hybrides combinant les deux méthodes sont également analysées.

Mots-clés : méta-heuristiques, grasp, recherches avec tabous, méthodes hybrides, transport public, systèmes adaptatifs de transport.

1 Introduction

Traditional public transport is evolving towards more flexible services and operation modes in order to better serve the needs of the population, capture additional demand, and increase its profitability. Traditionally, flexible services were designed for people with particular needs only (e.g., the *on-demand door-to-door transportation* for handicapped and elderly people [9, 21]). Recently, however, attempts are made to design transit services that may be adapted to a broader range of potential customers.

The transportation system that we contemplate integrates traditional bus transportation and *on-demand* services. In this case, transportation service is provided "as usual" among a limited number of specific points, which are served according to a predefined schedule and regardless of the actual requests. Other than these compulsory stops, customers can contact the transit company to request service between two given points and, in so doing, induce detours in the vehicle routes. Such a system attracts more passengers than a regular, fixed-route line, since "active" individual users are allowed to personalize itineraries, while still providing basic transportation to those who do not explicitly call for the service. Such a *Demand Adaptive System* (DAS) represents a good compromise between an expensive personalized service that precisely fulfills individual requests, and the cheap alternative supplied by the traditional public transport, that usually does not provide transportation along individual itineraries. It is a system that addresses the needs of relatively low demand areas (e.g., North American city suburbs) or periods (e.g., non-rush hours, evenings, week ends, etc.), as well as those cases where a basic transportation service must be guaranteed even though a traditional type of transit system is not economically worthy and efficiency becomes a critical issue.

Malucelli, Nonato, and Pallottino [20] proposed a mathematical formulation for a particular case of demand adaptive systems, the scheduling of one occurrence of a demand adaptive service line. Called DAS1, the formulation corresponds to the case of a number of small-to-medium European cities where transit companies operate a limited number of suburban lines in somewhat flexible mode. It also constitutes a major building block for more comprehensive DAS models where a large number of one-time-one-line sub-problems have to be solved repeatedly [6, 7, 3]. Thus, DAS1 requires very efficient solution methods.

The DAS1 model yields a mixed-integer formulation that is NP-Hard and that, for realistic dimensions, cannot be addressed by exact methods within acceptable computing times. Moreover, [20] clearly established the significant differences in problem and model structure between DAS and classical dial-a-ride problems. Consequently, the straightforward application of well-known vehicle routing problem algorithms does not appear as a viable approach to efficiently solving DAS problems. Thus, the challenge is to devise new approaches that, based on innovative as well as classical methodologies, exploit the problem structure and solve DAS1 formulations efficiently.

In this paper, we develop, test, and compare several solution strategies that belong to two general meta-heuristic classes: memory-enhanced greedy randomized multi-start con-

structive procedures and tabu search methods. For the former, we develop a framework where short and long term memories enhance the search. For the later, we study several neighbourhood-move strategies. The two solution approaches are then brought together to yield hybrid meta-heuristics that aim to capitalize on the respective strengths of each method. Careful algorithmic design facilitates the correct comparison of performances.

The contributions of this paper are as follows. First, we propose efficient solution strategies for a difficult combinatorial optimization formulation that is central to addressing an important new class of management and operation problems in intelligent public transportation. Second, two meta-heuristic strategies are fully developed and compared: a GRASP-based, memory-enhanced, constructive approach and several tabu search strategies. To the best knowledge of the authors, this is one of the rare efforts reported in the literature to design both the meta-heuristics and the experiments, including comparable best-efforts in coding the procedures, such that a thorough, fair, and enlightening comparison of the two meta-heuristics for a given problem class may be carried on. Finally, a number of hybrid approaches are also developed and analyzed.

The paper is organized as follows. DAS1 is formally described and the formulation is recalled in Section 2. Section 3 describes how to compute an upper bound via Lagrangian relaxation and introduces the general framework and basic components of the proposed meta-heuristics. Section 4 is dedicated to constructive GRASP-based heuristics, Section 5 investigates the implementation of improving tabu search meta-heuristics, while Section 6 describes three hybrid methods in which the constructive and improving strategies interact. Computational results are presented and analyzed in Section 7. Finally, conclusions and further directions of work are discussed in Section 8.

2 Problem Definition and Formulation

A *Demand Adaptive System* (DAS) may be designed as follows. Let us consider a set of lines: each traversal of a line is described, as usually, in terms of a set of time-tabled trips. The stops in the original time table are called *compulsory stops*. Such points are locations at which a demand for transportation may be naturally assumed to be present, such as, the stops of an express line, stops at rail or subway stations, or any other points which must be serviced according to the policies of the local transit authority. To introduce some flexibility into the vehicle routes, the vehicle is allowed to transit by each compulsory stop during a *time window*. Beside the compulsory stops, a set of stops to be activated on demand (*optional stops* hereafter) is available to the users. Between each pair of consecutive compulsory stops a set of optional stops is defined, which can be served by the vehicle on its way from a compulsory stop to the next. A user issues a *service request* specifying the pick up and drop off stops. In the absence of requests involving optional stops, the vehicle travels along the shortest path on the network within each pair of consecutive compulsory stops. The acceptance of a request implies the rerouting of the vehicle for that part of route involving the optional stop(s) related to the request. The detour may cause a delay of the transit time at the following stops. Traveling times of the arcs of the physical network are assumed known. Figure 1 illustrates a DAS line with its compulsory and optional stops, while Figure 2 displays a vehicle tour on the same line and a number of requests satisfied by that tour.

Several DAS models can be devised, depending on the policy used to deal with requests. In this paper, we focus on a specific case according to which requests are rejected if their acceptance causes the tour to become infeasible or is not economically worthy. Moreover, once a request has been accepted, the user must be boarded and alighted *exactly* at the desired stops. In the basic model, called DAS1 [20], a single line circuit is traversed once by a single vehicle and the most profitable subset of requests that can be feasibly served must be selected. This model may be readily generalized to several vehicles traversing the line and multiple intersecting lines with users traveling from stops of a line to stops of another line on vehicles connecting at compulsory stops [6, 3]. The solution strategies developed for DAS1 are the building blocks of more comprehensive formulations.

Consider a line structured as a circuit, served by a single vehicle, starting and ending its tour at the same terminal. Along the tour, the vehicle traverses a sequence $H = \{f_1, f_2, \ldots, f_{n+1}\}$ of $n+1$ compulsory stops according to a given order, where the terminal is the first (f_1) and the last ($f_{n+1} = f_1$) element of the sequence. For each stop f_h a time window $[a_h, b_h]$ is defined; the vehicle must leave f_h not before a_h and not later than b_h, but it may arrive there before a_h for $h = 1, \ldots, n$; b_{n+1} is the maximum trip completion time, and $a_1 = b_1$ is the starting time of the vehicle from the terminal.

A set F_h of optional stops is associated to each pair of consecutive compulsory stops $\langle f_h, f_{h+1} \rangle$. The vehicle passes by an optional stop only if a related boarding or alighting request at that stop has been issued. The sets F_h are mutually disjoint. Considering any pair $\langle f_h, f_{h+1} \rangle$ we can define a directed graph $G_h = (N_h, A_h)$, such that $N_h = F_h \bigcup \{f_h, f_{h+1}\}$

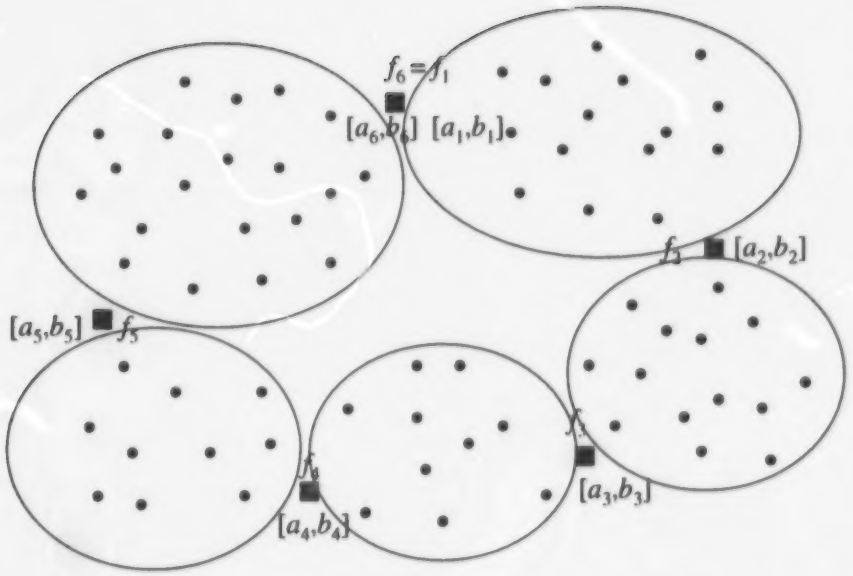


Figure 1: The Compulsory and Optional Stops of a DAS Line

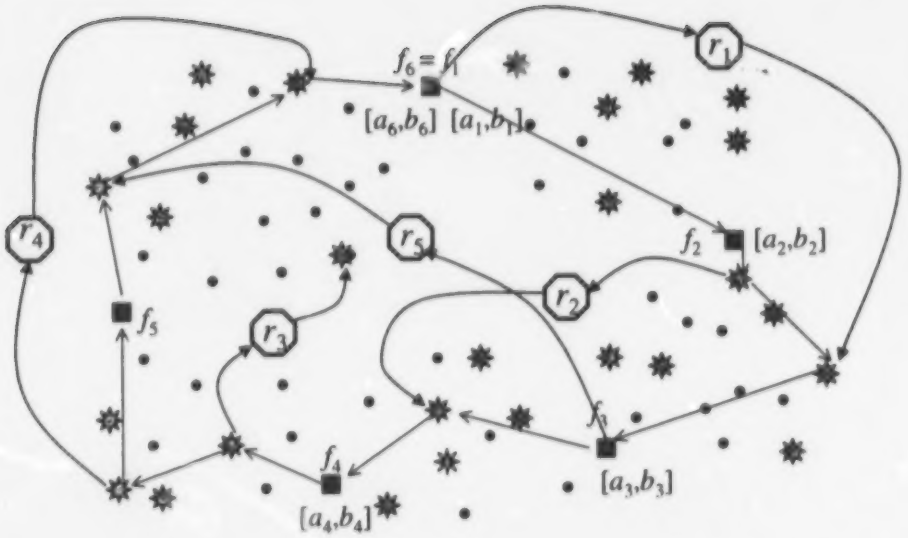


Figure 2: A Set of Requests and a Vehicle Tour on a DAS Line

is the stop set and $A_h \subseteq N_h \times N_h$ is the set of arcs connecting the stops. In the following, we shall refer to G_h as *segment* h. Finally, $G = (N, A)$ is the whole graph, where $G = \bigcup_h G_h$. The travel time τ_{ij} and the travel cost c_{ij}, for each $(i, j) \in A$ of possible consecutive stops, either compulsory or optional, are given and positive. Travel time τ_{ij} represents the duration of the shortest path from i to j and takes into account the time needed for stopping at i. Then the subgraph induced by F_h is a complete graph. Besides, triangular inequalities hold but time and cost matrices may be asymmetric.

Let us denote by P_h the set of paths in G_h from f_h to f_{h+1}. The vehicle itinerary in segment h is a path $p \in P_h$ whose travel time $\tau(p)$ and travel cost $c(p)$ is given by the sum of the travel times and the costs of its arcs, respectively:

$$\tau(p) = \sum_{(i,j)\in p} \tau_{ij}; \qquad c(p) = \sum_{(i,j)\in p} c_{ij}. \tag{1}$$

Let t_h be the starting time from f_h; without loss of generality, we assume $t_1 = a_1$. The sequence of paths defined for each segment forms a *tour* q starting and ending at the terminal. Let $p_h \in P_h$ be the path chosen in segment h. The arrival time at the end of the segment, i.e. at stop f_{h+1}, is then $t_h + \tau(p_h)$. The resulting tour q is *time-feasible* if:

(i) $t_h + \tau(p_h) \leq t_{h+1}$ $\qquad h = 1, \ldots, n-1;$

(ii) $t_n + \tau(p_n) \leq b_{n+1};$

(iii) $a_h \leq t_h \leq b_h$ $\qquad h = 2, \ldots, n.$

Since no time-feasible tour may contain a path in P_h whose travel time exceeds $b_{h+1} - a_h$, for all segments $h = 1, \ldots, n$, P_h can be restricted to the set of elementary paths from f_h to f_{h+1} with travel time less than or equal to $b_{h+1} - a_h$. To illustrate these notions, Figure 3 displays the same DAS line as before with the appropriate notation: five compulsory stops with time windows, four optional stops (this number has been reduced for a clearer figure), and nine paths (two each for segments 1, 2, 3, and 5, and one for segment 4).

Let us denote by R the *request set*, where request $r \in R$ is defined as a pair $\langle s(r), d(r)\rangle$ of boarding and alighting stops. Let $h(s(r))$ and $h(d(r))$ denote the boarding and alighting segments, respectively. We assume that, for each request $r \in R$ with both $s(r)$ and $d(r)$ optional stops, $h(s(r) < h(d(r))$ holds, i.e. $s(r)$ and $d(r)$ do not belong to the same segment. Such an assumption is quite realistic, since any two optional stops within the same segment are relatively close to each other. Because of this assumption, no precedence constraints must be enforced between stops within the same segment, while precedence constraints regarding the boarding and alighting stops of each request are implicitly handled by the sequencing of compulsory stops. Figure 3 illustrates four requests and the modifications each would induce to the vehicle route if accepted. Notice that request 3 has its boarding node at a compulsory stop (f_2), while request 2 alights at compulsory stop f_5.

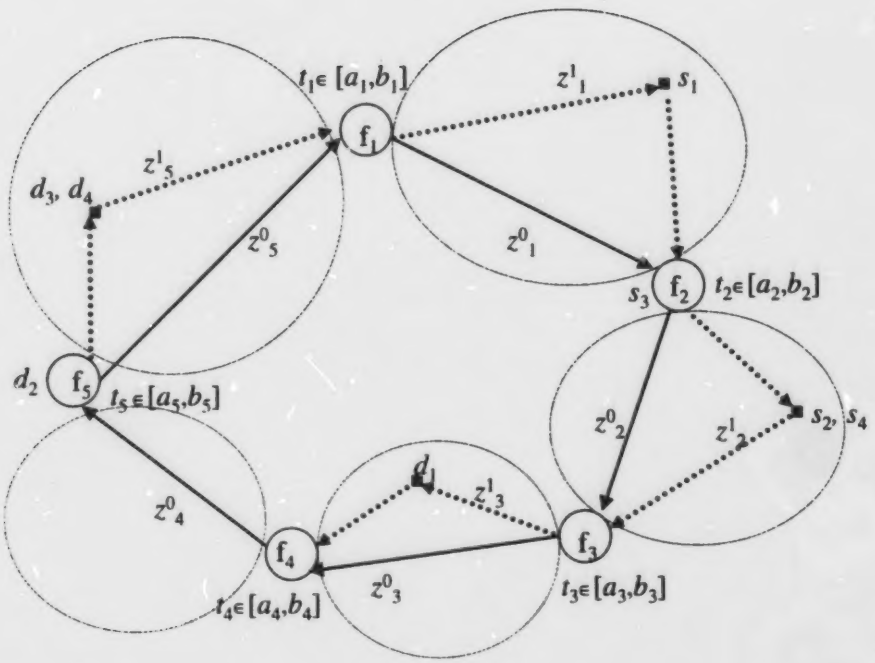


Figure 3: Paths of a DAS Line

Let $Q = \{q\}$ be the set of feasible tours; let $N(p_h)$, $h = 1, \ldots, n$, be the node set of path $p_h \in P_h$, i.e. the *served stops* in segment h, and let $N(q) = \bigcup_h N(p_h)$ be the set of stops served in tour q. According to model DAS1, given a tour q, a request r is *satisfied* if and only if both boarding and alighting stops belong to $N(q)$; then, the subset $R(q) \subseteq R$ of the requests satisfied by tour $q \in Q$ is given by:

$$R(q) = \{r \in R : s(r), d(r) \in N(q)\}.$$

A *benefit* $u(r) \geq 0$ is associated with each request $r \in R$. In its simplest form, the benefit corresponds to the fare charged for the particular ride, but it may include service-quality or priority indicators (e.g., a previously differed request must be served during the current run). The *global benefit* $u(q)$ of tour q is given by:

$$u(q) = \sum_{r \in R(q)} u(r).$$

The *global cost* $c(q)$ of tour $q \in Q$ is given by the sum of the costs of the paths forming q. Define the *profit* of a tour as the difference between benefit and cost. Then, the main objective of the service model DAS1 is to select the requests to be served and find a maximum profit feasible tour $q^* \in Q$:

$$u(q^*) - c(q^*) = \max\{u(q) - c(q) : q \in Q\}.$$

Define the following decision variables:

- $z_p^h = 1$ if path $p \in P_h$ is chosen, $z_p^h = 0$ otherwise, $\forall p \in P_h, h = 1, \ldots, n$ (see Figure 3);
- $y_r = 1$ if request r is satisfied, i.e. $s(r) \in N(p)$ if $z_p^{h(s(r))} = 1$, and $d(r) \in N(p')$ if $z_{p'}^{h(d(r))} = 1$, $y_r = 0$ otherwise, $\forall r \in R$;
- t_h is the starting time of the vehicle from f_h, $h = 1, \ldots, n$, with $t_1 = a_1$.

A path-based mathematical formulation of the DAS1 model may then be written as:

$$(\text{DAS1}): \; Z(y,z) = \max \sum_{r \in R} u(r) y_r - \sum_{h=1}^{n} \sum_{p \in P_h} c(p) z_p^h \tag{2}$$

$$y_r \leq \sum_{p \in P_h} \delta_{s(r),p} z_p^h \qquad \forall r : s(r) \in N_h, h = 1, \ldots, n \tag{3}$$

$$y_r \leq \sum_{p \in P_h} \delta_{d(r),p} z_p^h \qquad \forall r : d(r) \in N_h, h = 1, \ldots, n \tag{4}$$

$$\sum_{p \in P_h} z_p^h = 1 \qquad h = 1, \ldots, n \tag{5}$$

$$t_h + \sum_{p \in P_h} \tau(p) z_p^h \leq t_{h+1} \qquad h = 1, \ldots, n-1 \tag{6}$$

$$t_n + \sum_{p \in P_n} \tau(p) z_p^n \leq b_{n+1} \tag{7}$$

$$a_h \leq t_h \leq b_h \qquad h = 1, \ldots, n \tag{8}$$

$$y_r \in \{0,1\} \qquad \forall r \in R \tag{9}$$

$$z_p^h \in \{0,1\} \qquad \forall p \in P_h, h = 1, \ldots, n \tag{10}$$

where $\delta_{s(r),p}$ is a constant that takes value 1 if $s(r) \in N(p)$, and 0 otherwise, $\forall p \in P_{h(s(r))}$, $\forall r \in R$; analogously, $\delta_{d(r),p} = 1$ if $d(r) \in N(p)$, and 0 otherwise, $\forall p \in P_{h(s(r))}$, $\forall r \in R$.

A request r is served if and only if the vehicle passes by both its boarding $s(r)$ and alighting $d(r)$ stops. The tour is then *request-feasible* with respect to request r. Constraints (3) and (4) correspond to the linking constraints typical of network design formulations. They enforce the request-feasibility of the tour by linking the path choice to the requests served by the tour and by coupling boarding and alighting stops for each request. Constraints (5) impose the selection of one path for each segment, while constraints (6), (7) and (8) state the requirement that the selected paths form a time-feasible tour. Problem DAS1 is NP-Hard since the travelling salesman problem formulation reduces to a particular instance (one segment only).

DAS1 may be viewed as a particular dial-a-ride problem and classical heuristics for the later could be adapted for the former. In particular, constructive and improving heuristics could be devised that build a solution starting from a feasible tour. In the simplest constructive heuristics, for example, one may start from a tour visiting only compulsory stops and then, at each step, iteratively insert optional stops to satisfy more requests [22]. Similarly, improving heuristics can be devised based on exchanges of optional stops in and out of the current tour, as in classical tabu search algorithms for vehicle routing problems [16, 17, 12]. Indeed, except for column generation approaches [1], most dial-a-ride heuristics, either constructive or improving, are based on node or arc insertions, deletions, and exchanges. In a DAS1 framework, however, the computation depends on the solution at hand and not just on the specific nodes or arcs involved in the operation. Hence, the costs of node and arc-based operations are not uniquely defined and, depending on the current tour, the same move may have different evaluations. To build meta-heuristics based on such ill-defined moves is not correct algorithmic practice. Moreover, as shown in [20], DAS1 differs from dial-a-ride problems in many aspects: neither precedence nor capacity constraints are present, time windows are given at compulsory stops only, the stop set is partitioned into segments which must be visited in a predefined sequence but only a selection of stops is required to be visited, and so on. Thus, while DAS1 appears less constrained in the way nodes are sequenced within each segment, it requires the solution of a selection problem which involves pairs of nodes in different segments. This emphasizes the need for the development of specific algorithmic

tools that exploit the particular structure of the problem. This structure suggests working on paths rather than on single nodes and assembling and improving tours by swapping paths in and out of the current tour.

3 Solution Approaches

In this section, we describe how to efficiently compute an upper bound to the DAS1 formulation and introduce a general framework for the meta-heuristics discussed in the following sections.

3.1 Computing an Upper Bound via Lagrangian Relaxation

Formulation DAS1 has potentially an exponential number of design variables z_p^h. Therefore, even the solution of the continuous relaxation is a difficult task. In a column generation framework, for example, each column corresponds to an elementary path from a compulsory stop to the next and the subproblem aims to find the highest reduced cost column. This subproblem then reduces to the computation for each segment $h = 1, \ldots, n$ of an elementary maximum profit path no longer than a given duration $b_{h+1} - a_h$, where profit is computed with respect to the dual variables associated to constraints (3)-(7). Therefore, the subproblem itself is a difficult problem. Alternative formulations with a polynomial number of variables can be given, corresponding to arc-based formulations. It is known, however, that in order to avoid cycles within each segment, a time variable must be introduced for each node, such that the order relations between time variables of adjacent nodes along the selected path can be expressed as linear constraints by introducing a suitably big constant [8]. As a side effect, the linear relaxation of this formulation is by far looser than the linear relaxation of model DAS1.

A third approach to compute an upper bound is provided by the solution of the Lagrangian relaxation of model (DAS1). Let us apply Lagrangian relaxation to constraints (6) with multipliers μ, and Lagrangian decomposition to variables y_r with multipliers λ; call y_r^s variables y_r in the objective function and in constraints (3) and introduce a new variable set y_r^d to replace y_r in constraints (4), while adding and relaxing constraints (11) in Lagrangian fashion:

$$y_r^s = y_r^d, \quad \forall r \in R \tag{11}$$

This yields a set of separable subproblems, one for each segment h, of the form:

$$(P(h)_{\lambda,\mu}) : \max \sum_{r:s(r)\in N_h} (u(r) - \lambda_r) y_r^s + \sum_{r:d(r)\in N_h} \lambda_r y_r^d - (\mu_h - \mu_{h-1}) t_h - \sum_{p\in P_h} (c(p) + \mu_h \tau(p)) z_p^h$$

$$y_r^s \le \sum_{p\in P_h} \delta_{s(r),p} z_p^h \qquad \forall r : s(r) \in N_h$$

$$y_r^d \le \sum_{p\in P_h} \delta_{d(r),p} z_p^h \qquad \forall r : d(r) \in N_h$$

$$\sum_{p \in P_h} z_p^h = 1$$

$$a_h \leq t_h \leq b_h$$

$$y_r^s \in \{0,1\} \qquad \forall r : s(r) \in N_h$$

$$y_r^d \in \{0,1\} \qquad \forall r : d(r) \in N_h$$

$$z_p^h \in \{0,1\} \qquad \forall p \in P_h$$

Each subproblem $(P(h)_{\lambda,\mu})$ is of a smaller size than the global problem (DAS1) and, although still NP-Hard, is also more affordable to solve. In fact the request benefits (including the effect of multipliers λ) can be included in the weights of the nodes of graph G_h, and the costs of the links (including the effect of multipliers μ) can be included in the weights of the arcs of the same graph. The problem therefore reduces to a maximum path problem with a constraint on maximum travel time. Positive cycles may occur in graph G_h because of the weight definitions. But, due to the time windows constraints, the problem is bounded and can be solved as described in [8]. The additional constraint $t_h + \sum_{p \in P_h} \tau(p) z_p^h \leq b_{h+1}$, $h = 1, \ldots, n$, which is redundant in the original formulation, can be added to strengthen the bound without increasing the subproblem complexity [19].

Let $\Phi_h(\lambda, \mu)$ be the optimal value of the h^{th} subproblem with respect of the point (λ, μ). Then, the value of the Lagrangian function at (λ, μ) is given by $\Phi(\lambda, \mu) = \sum_{h=i}^{n} \Phi_h(\lambda, \mu)$. Since each evaluation of $\Phi_h(\lambda, \mu)$ yields a path from h to $h+1$, the set of the n paths associated with the solution of the n subproblems yields a tour, which in general is not time-feasible, for each evaluation of the Lagrangian function. The Lagrangian Dual, i.e. the problem of minimizing $\Phi(\lambda, \mu)$, is solved by way of a bundle algorithm [2] which provides a vector of optimal multipliers (λ^*, μ^*), a collection of feasible paths for each segment (a path is generated at each iteration of the bundle algorithm), and an upper bound to the problem given by $\Phi(\lambda^*, \mu^*)$. Notice that these paths, called *promising paths* hereafter, are closely related to the continuous optimal solution of DAS1. In bundle algorithms, optimality conditions for a point (λ^*, μ^*) are stated with respect to the minimum norm convex combination of a bundle of ε-subgradients of the point. Let $SG^* = \{g_b\}$ be the set of ε-subgradients at (λ^*, μ^*) with non-zero multiplier in such a combination. From duality theory, it is known that, provided (λ^*, μ^*) is optimal, the optimal solution to the linear relaxation of DAS1 can be obtained from the convex combination of the solutions associated with each $g_b \in SG^*$. The promising paths are indeed those computed during the evaluations of the Lagrangian function that yielded the ε-subgradients in SG^*.

The collection of promising paths provides a good starting base for heuristic methods that build and improve tours by assembling paths, one for each segment, such as the *path-based meta-heuristics* that are introduced in the next section.

3.2 A general framework for path-based meta-heuristics

Two meta-heuristic approaches are discussed in this paper, *memory-enhanced randomized multi-trial* heuristics and *tabu search* procedures. The former belongs to the constructive class of meta-heuristics and it raises the issue whether and how it is possible to define a measure of the potential benefit of a path in a framework where only a partial solution is known. This leads to the to the need to add long term memories to initial the GRASP-like [11] procedure. The later is a classical tabu search heuristic [13, 14, 15] based on the use of memory as a mean of recognizing good solutions and using history on "good" solution attributes to guide the search.

Despite their fundamental difference in the search design, both methodologies share a number of elements, particularly the basic neighbourhood and move definitions. Moreover, when developing the best search strategy for each methodology, we aimed at using similar concepts and structures, as illustrated by the advanced move evaluation function presented later on. This, of course, makes good software engineering sense by facilitating the utilization of the same code parts in the various programs. Moreover, it also allows us to conduct a correct evaluation and comparison of the two methodologies, as well as easily build and experiment hybrid meta-heuristics that attempt to combine the best strategies of each.

In the following, we introduce the basic concepts and components which are common to all the proposed algorithms. Each will be later detailed according to the specific context.

3.2.1 Neighbourhoods and moves

Both meta-heuristic approaches are built around a local search phase that iteratively attempts to improve a current solution and build a maximum profit tour. The core of the algorithm thus consists in swapping paths in and out of a tour. Starting from a feasible solution, at each step the current tour is modified by selecting one path not in the tour from a given set of candidate paths, and swapping it with the path currently in the tour in the corresponding segment. As already indicated, the set of promising paths provides the initial *pool* of candidate paths, even though such an initial set could be generated by any other method as well.

Formally, define

- A pool of paths $P' = \bigcup_h P'_h$, $P'_h \subseteq P_h$, where P'_h corresponds to the set of paths already generated for segment h;
- The *basic path* of segment h, denoted $\bar{p}_h, h = 1, \ldots, n$, as the minimum travel time path from f_h to f_{h+1} without intermediate stops;
- The *basic tour* $\bar{q}$ formed by the n basic paths;
- The tour q as the current time-feasible solution and $p_h(q)$ the path of q in segment h.

To illustrate, consider the line of Figure 3. The line is made up of five segments and the figure displays the basic paths for each segment and the corresponding basic tour. The requests associated to this example induce four other paths. Assuming that accepting all requests is feasible with respect to time, the current tour includes alternate paths for segment 1, 2, 3, and 5, and the basic path for segment 4.

Initially, P' contains the basic and promising paths. Then, at each step, the *best* path in P' with respect to q for a given segment h is selected according to an evaluation function: the *score*. A new tour may then be built from q by *swapping* the selected path with $p_h(q)$, if the new tour is still time-feasible; the next best path is selected otherwise. This is the main search principle of the tabu search meta-heuristic. By the same approach, the constructive heuristic gradually replaces the basic paths of the initial solution to build a good feasible solution.

The path swapping operation thus represents the fundamental move of the path-based meta-heuristics we introduce. Then, the neighbourhood of the current solution q is the set of all time feasible tours which can be obtained by way of a single path swap move, while the pool P' defines the restricted neighbourhood on which swaps are actually evaluated and performed.

Two main issues must be addressed in order to define and implement meta-heuristics based on this swap move: how to verify the time-feasibility of the tour yielded by the swap and the criteria according to which candidate paths are selected. Moreover, given a feasible tour, "post-optimization" steps have generally to be taken to insert, delete or reorder nodes. Although these operations concern the paths in the tour only, they also provide opportunities for the generation of new paths, thus enriching the pool for subsequent iterations. This *dynamic* pool mechanism, where the initial pool of promising paths evolves along with the search trajectory, provides the means to broaden the scope of the search.

3.2.2 Time feasibility

To check the time-feasibility of a new tour q, observe that each path $p_h(q)$ can be thoroughly characterized by its earliest and latest departure time from f_h, EDT and LDT, respectively. These values can be iteratively computed according to:

$$EDT(p_h(q)) = \begin{cases} a_1 & \text{if } h = 1, \\ \max\{a_h, EDT(p_{h-1}(q)) + \tau(p_{h-1}(q))\} & \text{if } h = 2, \ldots, n; \end{cases}$$

$$LDT(p_h(q)) = \begin{cases} \min\{b_n, b_{n+1} - \tau(p_n(q))\}; & \text{if } h = n; \\ \min\{b_h, LDT(p_{h+1}(q)) - \tau(p_h(q))\} & \text{if } h = n-1, \ldots, 1, \end{cases}$$

The earliest and latest arrival time at f_{h+1}, EAT and LAT, respectively, may then be directly derived from the two previous values:

$$EAT(p_h(q)) = EDT(p_h(q)) + \tau(p_h(q)) \qquad h = 1, \ldots, n; \tag{12}$$

$$LAT(p_h(q)) = LDT(p_h(q)) + \tau(p_h(q)) \qquad h = 1, \ldots, n. \tag{13}$$

Let us define τ_h, the maximum travel time for paths in P'_h in order to qualify for swapping:

$$\tau_h = LDT(p_{h+1}(q)) - \max\{EAT(p_{h-1}(q)), a_h\}.$$

Then, path $p \in P'_h$ can be swapped with $p_h(q)$ only if $\tau(p) \leq \tau_h$.

3.2.3 Path selection criteria

Let p and p' be two paths in two different segments of tour q. Define $\theta(p, p')$, the *mutual benefit* of paths p and p', as the sum of the benefits of all requests with boarding node in p or alighting node in p' and vice-versa. One may then state the profit of tour q in terms of its paths and the objective function of DAS1 as the sum of the mutual benefit for each pair of paths in the tour, minus their costs:

$$u(q) - c(q) = \sum_{h=1}^{n}(-c(p_h(q)) + \sum_{l<h} \theta(p_h(q), p_l(q))). \tag{14}$$

To select the best path swap move, paths $p \in P'$ may be ranked according to a score $s(p)$ that is a measure of the benefit coming from including path p in the tour. The contribution of a path to the cost of a solution is not invariant, however. Rather, it depends on the other paths in the tour, as emphasized by the coupling constraints (3)-(4) and equation (14). Thus, for example, the benefit of a request whose alighting node is in p contributes to the value of the solution only provided that the final tour passes by the boarding node (the same, conversely, applies to requests with boarding node in p). Yet, when a particular swap move is contemplated, one does not know the paths that will make up the final tour. Even for the last move (assuming one could identify this situation), this knowledge is not available since the tour may be modified further by any one of the "post-optimization" operations introduced in the next subsection. Therefore, the strategy that appears most appropriate in these circumstances consists in building selection criteria that approximate the true benefit of a path p to the final tour by using a path score function that varies according to the current solution q.

Setting up a meaningful score function represents a challenging issue within the framework of a constructive heuristic, where at the time of the path selection the final tour is only partially determined. In particular, founding such an approximation on the evaluation of the mutual benefit of the paths in the pool, would be extremely expensive and not robust. A more viable alternative, used in several meta-heuristics in case of incomplete information, is to learn by experience. Section 4 describes such an implementation of the score function in the contest of a multi-trial constructive heuristic with learning capabilities. The operation appears significantly more straightforward within an improving heuristic where the other paths in the current tour are known and the profit of the modified tour can be computed

exactly based on equation (14). This direct approach still misses the tour modification that could follow from a post-optimization operation, however. Thus, we discuss in Section 5 how some of the concepts developed in the context of the constructive heuristic may be used to enhance the score used in the tabu search method that we propose.

3.2.4 A dynamic population of paths

The pool of paths evolves dynamically during the search: new paths are added to the pool, others are modified, some may be discarded. Changes in the path set occur due to generation of paths by node deletion or insertion, extraction of sub-paths from time-infeasible paths, path optimization by node reordering, path deletion due to pool overflow. We now briefly address each of these issues.

Node deletion

Recall that a request r is defined by two nodes $s(r)$ and $d(r)$ that identify its boarding and alighting stops, respectively. We denote node i as the *mate* of j with respect to request r if either $i = s(r)$ and $j = d(r)$ or vice-versa. Not all requests associated to a node of a given tour are necessarily served by that tour. Thus, for example, when paths are swapped, the resulting tour is generally no longer feasible with respect to the requests associated to the nodes it serves. Indeed some requests previously served may have lost their boarding or alighting node. Similarly, some of the requests associated to the nodes of the entering path may not find their mate nodes in the paths of the other segments. Such requests are called *orphans* with respect to the current tour and are not associated with it. A *profitless* node with respect to a given tour serves only orphan requests. Such a node yields no profit and increases the cost and travel time of the tour. It should therefore be discarded. Clearing paths of profitless nodes in the current tour, possibly originates new paths thus enriching the pool. Nevertheless, since this information is local to the current solution, the original paths are kept in the pool.

Path reduction

Paths may also be generated during pool inspection, when searching for the best candidate path for swapping. If the travel time of a path is longer than required by time-feasibility, any sub-path extracted from this path by the deletion of some optional stops could be considered as a candidate for swapping. Let p be the selected path in P'_h. We rank the nodes in $N(p) \cap F_h$ according to the ratio of the gain in time over the loss in revenue due to bypassing the node, where the gain in time for node i (δ_i^t) is defined as

$$\delta_i^t = \tau(p) - max\{\tau(p^{i-}), a_{h+1} - b_h\}, \tag{15}$$

where $\tau(p^{i-})$ is the duration of path p after the removal of node i; note that δ_i^t it is strictly positive since $\tau(p) > \tau_h \geq a_{h+1} - b_h$ holds. The loss in revenue $\delta_i^{\tau v}$ is computed according to the specific implementation of the score function. The node with the maximum value is

then selected and deleted. The resulting path is added to the pool for use in subsequent iterations.

Node insertion

Tours are modified by various node deletion or reordering (see below) operations resulting in extra time for additional detours. The node insertion or *mending* step attempts to locally improve the solution by inserting additional nodes to increase the number of requests served and, therefore, the total benefit of the tour. Nodes are inserted sequentially, according to a criterion which depends on the specific implementation. Indeed, even in the context of a constructive heuristic the contribution of each node to the current solution value can now be computed exactly, since a complete tour is given; however, this is still local information which does not take into account the mutual interaction of the additional nodes. Alternative criteria that try to capture the full potential of nodes are described together with the general solution meta-heuristics.

Node reordering

In a post-optimization-like phase, paths may be modified regarding the order nodes are traversed, to improve the overall efficiency by decreasing the path duration. In fact, the ordering was computed initially during the solution of the Lagrangian Dual on the basis of reduced costs. However, successive insertions and deletions may have considerably deteriorated the sequencing. Identifying the optimal node sequence in a segment belongs to the TSP family [18, 16]. Due to the difficulty of the problem, paths are checked only for local optimality with respect to the 2-opt neighborhood defined by the exchange of a single pair of arcs. Intuition and experimental evidence suggests to locally optimize paths as soon as they enter a solution. Efficiency concerns alter this policy as indicated in the following sections.

Pool management

Creation of new paths may cause pool overflow (a maximum size of the pool is given at the beginning). A pool management strategy is therefore needed. This strategy addresses issues such as the frequency of path creation by, for example, systematic application of path reductions, the path ranking criterion used to discard paths deemed no longer interesting, the number of paths discarded, the frequency of the re-initialization of the pool (if any). Pool management strategies are specific to each meta-heuristic class and are described in the corresponding sections. A general comment can be however made concerning the ranking of paths in the pool for possible deletion.

The goal is to keep a population of paths that may contribute to obtaining a better solution through construction or improving techniques. This implies considering path impacts over a longer horizon that the immediate contribution to a tour. The ranking criterion should somehow reflect this concern. In meta-heuristic search, past history is often used to guide the search or to guess the future value of a given element. These learning capabilities are based on various memory mechanisms about the behaviour of solution components or

attributes with respect to "good" or improving solutions and should be put to good use in developing a ranking function for pool management. Tabu search methods are designed to construct and use these mechanisms - one may say that memories and learning are at the core of tabu search. This is not true of classical multi-start heuristics. This is one additional reason that explains why we have enhanced our multi-start constructive heuristics with a number of long term memories as described in the next section.

4 Memory-enhanced Constructive Heuristics

The constructive class of meta-heuristics that we consider is inspired by the multi-trial randomized heuristics, called GRASP by its authors [11], that have shown good behaviour on related problems (e.g., maximum independent set [11]). Based on the neighbourhood and path swapping operation defined previously (Section 3.2.1), the basic multi-trial greedy heuristic iterates as follows:

1. Start with the basic tour $\bar{q}$ as the initial solution.
2. **Outer loop**: Perform *max_iter macro-iterations*. At each macro-iteration, a feasible tour is constructed by performing (at most) n successful swap moves, one for each segment (the segment loop). At the end of the macro-iteration, a local improvement of the current tour is performed by re-ordering the nodes of the selected paths and by inserting new nodes according to a specific priority function.
3. **Segment loop**: Randomly select a segment not yet examined and identify the best path in the pool to be swapped with the basic path such that the tour continues to be time-feasible (the path loop). Following the move, *sure* orphan requests and profitless nodes are dropped and the pool is updated by inserting the new paths generated during the path loop.
4. **Path loop**: All paths in the pool for the given segment are evaluated and sorted according to the score function. Paths are then examined in decreasing score order until either a time-feasible swap is found or all paths are examined. In the latter case (*null swap*), the basic path will belong to the solution of the current macro-iteration, leading to the rejection of all requests involving an optional stop of that segment. Path reduction is applied to each path in the pool that does not offer a time-feasible swap.

The fundamental issues that need to be addressed in order to completely characterize the methodology concern the definition of a meaningful score function and the randomization strategy.

4.1 Score functions

A straightforward score function based on (14) can be defined as follows. Given a partial solution q, let H^+ be the index set of all segments for which a path has already been selected, so called *examined* segments, and let $H^- = H \setminus H^+$ be its complement, i.e. the index set of *open* segments. Given $p \in P'_h$, for each segment $l \in H^- \setminus h$, the path p_l^* such that $\tau(p_l^*) \leq \tau_l$ and which maximizes $\theta(p, p_l^*)$ is considered. The score $s(p)$ is defined as in (16),

$$s(p) = \frac{1}{2} \cdot \sum_{l \in H^- / (h)} \theta(p, p_l^*) + \frac{1}{2} \cdot \sum_{h \in H^+} \theta(p, p_h(q)) - c(p), \tag{16}$$

and it depends on the path cost, the mutual benefits of p and the selected paths, and the mutual benefits of p and p_l^* for the open segments. Note that the tour given by the selected path p and paths p_l^* may not be time-feasible. Hence, this score is an upper bound on the contribution, according to (14), of p to the profit of any tour containing p and the paths already selected. It represents a "most optimistic" evaluation with respect to what is still unknown (the future paths belonging to the tour).

This approach did not perform well, however. First, the initial computation of the score and its continuing updating is computationally expensive since it depends on the size of the pool. Moreover, a path already in the tour may become time-infeasible due to the selection of other paths in adjacent segments. Second, the outcome is highly dependent on the sequence according to which segments are processed. In particular, requests involving stops in segments that are processed first have higher chances to be satisfied. Thus, the final outcome is biased.

In order to overcome these limits, the method has been enhanced with a number of additional features that allow the development of a more meaningful and computationally-tractable score function. In particular, we try to build up knowledge on the *potential performance* of each path and node based on the solutions which have been visited previously during the search process. This is naturally implemented by way of long-term frequency and recency memories. The main idea is to approximate the value of a path by estimating the revenue that may be gained by the satisfaction of requests boarding or alighting at the nodes of the path. This estimation is to be carried out by evaluating the probability of satisfying requests based on the record of the outcome of the previous macro-iterations.

The contribution of a node i to the final solution depends on the selection of its mates. Therefore, given r such that $i = s(r)$ or $i = d(r)$ and the mate of node i has been neither included in nor discarded from the tour being built in the current macro-iteration, we are interested in the probability of satisfying r once that i has been selected.

Define $P_s(r)$ ($P_d(r)$), the probability to satisfy request r given that node $s(r)$ ($d(r)$) has been selected and its mate is not discarded. The *potential revenue* of node i, $npr(i)$, may then be defined as the sum of the revenue on all requests r such that i is either $s(r)$ or $d(r)$, multiplied by the appropriate approximations of these probabilities:

$$npr(i) = \sum_{r:i=s(r)} u(r)\widetilde{P}_s(r) + \sum_{r:i=d(r)} u(r)\widetilde{P}_d(r). \tag{17}$$

The node potential revenue may be used for "long" term decision concerning nodes such as, for example, in the path reduction operations where in order to select the node to be removed, one has to estimate the loss in revenue caused by the deletion of a node. Symmetrically, node potential revenues are used to select nodes to be inserted in the tour mending operation of the tabu search procedure.

Potential node revenues are static information with respect to a single macro-iteration since they do not take into account the current partial solution. That is, these measures do

not account for the fact that when a candidate path is being selected, some of the mates of its nodes might actually have already been processed in the current macro-iteration. We therefore introduce a *dynamic* estimation of the potential benefit of a node with respect to the current tour, $dnpr(i,q)$. This value is initialized at the beginning of each macro-iteration to the value of the static measure (17), but it is then updated as a function of the partial tour:

$$dnpr(i,q) = \sum_{h \notin H^+} \left(\sum_{r:s(r)=i \wedge d(r) \in F_h} u(r)\widetilde{P}_s(r) + \sum_{r:d(r)=i \wedge s(r) \in F_h} u(r)\widetilde{P}_d(r) \right) + \sum_{r \in N(i,q)} u(r) \quad (18)$$

where $N(i,q) = \{r : s(r) = i \wedge d(r) \in N(q)\} \bigcup \{r : d(r) = i \wedge s(r) \in N(q)\}$ represents the set of requests with one end-node in the selected path and the other in one of the other path already in the tour.

At the end of each macro iteration, equation (18) yields the exact evaluation of the additional benefits coming from the insertion of node i in the current tour q. $dnpr(i,q)$ is thus used in node selection during the request feasibility mending process in post-optimization phase. Moreover, based on $dnpr(i,q)$, we may define the *dynamic score* function of a path

$$dppr(p,q) = \sum_{i \in N(p)} dnpr(i,q), \quad (19)$$

as well as the *score* function $ds(p)$ defined as the ratio of the difference between the dynamic potential revenue and the path cost, and the time used by the path:

$$ds(p,q) = \frac{dppr(p,q) - c(p)}{max\{\tau(p), a_{h+1} - b_h\}}. \quad (20)$$

To estimate the request satisfaction probabilities $P_s(r)$ and $P_d(r)$, one may record the number of favorable cases (i.e., number of times i has been selected before its mate was examined) and possible cases (i.e., number of times the selection of i has been followed by the selection of its mate). To this purpose, for each request r, we record the number of times node $s(r)$ has been selected before $d(r)$ was processed, and the number of times the selection of $d(r)$ actually followed, thus satisfying the request. Let us call these *promising* and *realized* (*held*) *memories* $\mathcal{S}^{pr}(r)$ and $\mathcal{S}^{hd}(r)$, respectively. The same information is also kept for the boarding node of each request $d(r)$: $\mathcal{D}^{pr}(r)$ and $\mathcal{D}^{hd}(r)$, respectively. These counters are updated at the end of each macro-iteration. The probabilities then become simply:

$$\widetilde{P}_s(r) = \frac{\mathcal{S}^{hd}(r)}{\mathcal{S}^{pr}(r)} \quad \text{and} \quad \widetilde{P}_d(r) = \frac{\mathcal{D}^{hd}(r)}{\mathcal{D}^{pr}(r)} \quad (21)$$

4.2 Randomization and memories

The score function (20) is used for path selection purposes. However, it is further enriched by a number of randomization terms that impact the relative evaluations of nodes and paths.

Moreover, memories are introduced to record, on the one hand, the long-term evolution of the solution values generated by the constructive heuristic and, on the other hand, the performance of nodes and paths in the most recent macro-iterations. The information gathered in these memories is then combined to the randomization factors to further bias the node and path revenues and, thus, the selection process of the constructive meta-heuristic.

The idea behind the memory-based perturbations is that, when a reasonably good solution is found, it should be worked upon, trying to improve it, before eventually discarding its paths. Consequently, one aims to increase the likelihood of selecting paths that have belonged to solutions with the high scores while, at the same time, discourage the selection of paths that lately have performed poorly. The random perturbations play the classical role in grasp-like procedures of a diversification mechanism to broaden the search scope and escape local optima, as described in the following.

Two main mechanisms are implemented: *bonuses* for node and path revenue evaluations and the *smoothing* of the estimations of the request satisfaction probabilities.

Let $\rho(p)$ and $\rho(i)$ denote the perturbation factors associated to path p and node i, respectively. $\rho(i)$ is added to the node probabilistic revenue, while $\rho(p)$ is added to the path score. The relation (19) then becomes:

$$dppr(p,q) = \rho(p) \sum_{i \in N(p)} \rho(i)\ dnpr(i,q), \tag{22}$$

The perturbation factors are computed at the beginning of each macro iteration and are made up of two multiplicative terms: a random element (specific for each node and path) and a bonus. The bonus value is updated according to how good the solution found by the last macro iteration is compared to all previous solutions. The update proceeds as follows. The solution of the last macro iteration is compared to all previous ones. Then, the nodes and paths that make up the solution receive a bonus factor selected in a $[1-\beta,\ 1+\beta]$ interval, proportionally to the solution rank (a value of 1 is used for all paths and nodes not part of that solution). The β coefficient has to be calibrated.

The second mechanism concerns the estimation of the request satisfaction probabilities $\widetilde{P_s}(r)$ and $\widetilde{P_d}(r)$. The idea is to introduce into the current evaluation a measure of the values used in previous iterations. To that effect, the probabilities used during the last iteration, $\widetilde{P_s}^{k-1}(r)$ and $\widetilde{P_d}^{k-1}(r)$, are kept, the $\mathcal{S}^{hd}(r)$, $\mathcal{S}^{pr}(r)$, $\mathcal{D}^{hd}(r)$, and $\mathcal{D}^{pr}(r)$ memories are updated, and the probabilities at the current macro-iteration k are computed as the convex combination of the old and new evaluations:

$$\widetilde{P_s}^{k}(r) = \alpha \widetilde{P_s}^{k-1}(r) + (1-\alpha)\frac{\mathcal{S}^{hd}(r)}{\mathcal{S}^{pr}(r)} \tag{23}$$

$$\widetilde{P_d}^{k}(r) = \alpha \widetilde{P_d}^{k-1}(r) + (1-\alpha)\frac{\mathcal{D}^{hd}(r)}{\mathcal{D}^{pr}(r)} \tag{24}$$

This results in a smoothing effect that reduces the impact of the randomly selecting

segments at each macro-iteration and stabilizes the search process. The experimentation performed to calibrate the procedure has shown that all $\rho(p)$, $\rho(i)$, and α parameters are required for good performance of the constructive heuristic (see Section 7).

The final memory-enhanced, randomized, multi-trial constructive heuristic is as follows:

Initialization

- $q_0 = q_{best} = \bar{q}$; $Z_0 = Z_{best} = \bar{Z}$; $k = 0$; $\mathcal{Z} = \{z_0\}$.
- $\mathcal{S}^{hd}(r) = \mathcal{S}^{pr}(r) = \mathcal{D}^{hd}(r) = \mathcal{D}^{pr}(r) = 0$; $\widetilde{P}_s^0(r) = \widetilde{P}_d^0(r) = \frac{1}{2}$
- $P' = \{$Basic (in $\bar{q}$) and promising (generated during the computation of the Lagrangian Relaxation upper bound) paths $\}$.

Macro-iteration $k = k + 1$

1. $q_0 = \bar{q}$; $Z_0 = \bar{Z}$;
2. Evaluate $\widetilde{P}_s^k(r)$ and $\widetilde{P}_d^k(r)$ using relations (24)
   (except for the first 10 iterations for which $\widetilde{P}_s^0(r) = \widetilde{P}_d^0(r) = \frac{1}{2}$).
3. Randomly select a segment h.
4. Rank the paths for segment h in decreasing order according to the dynamic score function (20) and examine them until a time-feasible swap is found:
   - If an examined path does not lead to a time-feasible swap, apply path-reduction and update the pool;
   - If no time-feasible swap exists, the basic path will be part of the final tour.
5. Swap paths in segment h to obtain a new tour q_s^k;
   Drop orphan requests to make q_s^k request-feasible;
   Eliminate all profitless nodes. Update the pool.
6. Post-optimize the tour and update the pool.
7. Update Z_{best} if necessary. $\mathcal{Z} = \mathcal{Z} \bigcup Z_q$.
8. Update memories $\mathcal{S}^{hd}(r)$, $\mathcal{S}^{pr}(r)$, $\mathcal{D}^{hd}(r)$, and $\mathcal{D}^{pr}(r)$; Update $dnpr(i, q)$.
9. If stopping criterion not verified, goto 1.

5 Improving Heuristics: Tabu Search Procedures

Tabu search meta-heuristics [13, 14, 15] have consistently offered efficient and high quality solution methods to complex and hard combinatorial formulations, including vehicle routing [16, 17] and network design [4, 5] problems that display many similarities with the issues we address in this paper. The tabu search family of algorithms described in this section makes use of the same concepts - neighborhoods and moves, evaluation mechanisms, solution enhancement methods, and pool management strategies - introduced in Section 3 and implemented in the memory-enhanced multi-trial procedures of the previous section. In addition, it makes use of the learning capabilities of tabu search to efficiently guide the search towards not yet explored zones of the solution space.

The method is an improving meta-heuristic that explores the solution space of time and request-feasible tours. Given such a tour q and one of its segments, a different tour may be obtained by a move that swaps the path in the tour and a path outside the tour (in the pool). The *swap move* makes sure that the resulting tour is still time-feasible. It generally results, however, in some requests having only their boarding or alighting stops in the tour. These orphan requests are dropped from the tour to re-establish its request-feasibility, as are profitless nodes (not currently serving any request) requests. Following this *cleaning*, a *mending* operation is then called upon to eventually add requests to the tour.

We define three *neighbourhoods* that to be explored by using the basic swap move. The differences follow from the accessibility restrictions one applies to paths in various phases of the tabu search procedure:

- *Local Exploration* (*Neighbourhood Search* phase): The set of all time-feasible tours obtained by a swap move, followed by path cleaning and mending operations to yield a request-feasible tour.
- *Intensification*: The set of feasible tours that include requests often part of the best tour during the current Neighbourhood Search phase.
- *Diversification*: The set of all feasible tours except those that include paths often encountered in the best global solution.

Other that the $\mathcal{S}^{hd}(r)$, $\mathcal{S}^{pr}(r)$, $\mathcal{D}^{hd}(r)$, and $\mathcal{D}^{pr}(r)$ memories required to compute the node potentials (Section 4), three *memories* are defined to guide the search through these three neighbourhoods:

- $\mathcal{T}(p)$: Short term tabu memory. Implemented as a tag chosen randomly in a [*tabu_min*, *tabu_max*] interval, it indicates the iteration path p may be considered again for inclusion in the tour. $\mathcal{T}(p) = 0$ implies the path is not tabu.
- $\mathcal{F}^{\mathcal{Q}}(p)$: Path *frequency* memory - for how many iterations path p has appeared in the best global tour.

- $\mathcal{RF}^{\mathcal{L}}(r)$: Request *frequency* memory in the current Neighborhood Search phase - for how many iterations request r has been served by the best local tour.

The basic tabu search procedure follows then the classical form:

1. A **Neighbourhood Search** (the "local" search phase) is performed until a given number of unimproving consecutive iterations are observed. At each iteration, a segment is first selected. The non-tabu path in the pool that ensures the best swap with the path currently in the segment is then identified, the move is implemented, and the memories and best solutions are updated. Cleaning and mending operations complete the tasks performed during an iteration.

2. If the local best solution has been improved during the last Neighbourhood Search phase, an **Intensification** phase is performed to attempt to further improve it. The basic principle is to "lock" in the solution the elements that appeared most often in the best local solution. However, building this phase on paths may rapidly lead to situations where no paths may be moved out of or in the solution. We therefore work with requests: we forbid paths that do not contain requests that have appeared most often in the best local solution to enter the tour for a certain *intensification tabu tenure* *tab_int*, and start a new Neighbourhood Search phase.

3. If the local best solution has not been improved during the last Neighbourhood Search phase, we **Diversify** the search. In this phase, a number of paths that have most often appeared in the best global tour are swapped out of the solution and a tabu tag forbids their re-entry for a number of iterations; This constructs the set $\mathcal{D}$. A new Neighbourhood Search phase is initiated from the resulting tour.

To present the procedure in detail, recall that (q, Z) and $(\bar{q}, \bar{Z})$ represent the current and basic (no optional stops) tours, and their corresponding objective values, respectively, and define

- q_{best}, Z_{best}: Overall, global, best tour and its corresponding objective value.
- q_{local}, Z_{local}: Best local tour during the current Neighborhood Search phase and its corresponding objective value.
- k: Iteration index of a Neighborhood Search phase.
- i: Number of consecutive Neighborhood Search iterations without improvement.
- div: Diversification index.
- $s(p_h, q)$: *Score* associated to introducing path $p \in P'_h$ into tour q.
- Δ_{local}: Maximum permitted number of consecutive Neighborhood Search iterations without improvement.

- *tab_int*: Maximum length of tabu tenure following an Intensification phase.
- *N_int*: Number of requests considered in an Intensification phase.
- *tab_div*: Maximum length of tabu tenure at the beginning of a Diversification phase.
- *N_div*: Number of paths swapped out to initiate a Diversification phase.
- *Max_div*: Maximum number of diversifications.

The basic framework of the tabu search procedure we propose may then be written as follows:

Initialization

- $q_{local} = q_{best} = \bar{q}$; $Z_{local} = Z_{best} = \bar{Z}$; $k = 0$, $i = 0$, $div = 0$;
- $P' = \{$Basic (in $\bar{q}$) and promising (generated during the computation of the Lagrangian Relaxation upper bound) paths $\}$.

Neighborhood Search (the $k = k + 1$ "local" search phase)

1. Randomly select a segment h.
2. Select a non-tabu path $p_h^k \in P'$ such that it is time-feasible and has the best *score*. An *aspiration criterion* is used: if p_h^k is tabu but the swap would improve Z_{best}, the move is selected anyway.
3. Swap paths in segment h to obtain a new tour q_s^k;
   Assign tabu tenure to the exiting path;
   Drop orphan requests to make q_s^k request-feasible and eliminate all profitless nodes;
   Update the pool.
4. Try to *mend* the tour as described bellow; Let q_m^k be the new tour.
5. *Select* q_m^k or q_s^k as the new current solution q^k.
6. Update pool with paths of q_m^k; Update Z_{local} and Z_{best} if necessary;
   Update long-term memories: $\mathcal{F}^{\mathcal{G}}(p)$ (frequency in best global solution) and $\mathcal{RF}^{\mathcal{L}}(r)$ (frequency in best local solution).
7. If Z_{local} not improved, $i = i + 1$, otherwise $i = 0$;
   If $i \leq \Delta_{local}$ goto 1; Otherwise, perform an **Intensification** phase: For *N_int* requests
   - Select the request not yet considered with the highest value in $\mathcal{RF}^{\mathcal{L}}(r)$;
   - Attempt to insert the request in all the paths in the pool that belong to the segments corresponding to the boarding and alighting nodes of the request (this includes the paths currently in the tour);

- For the same two segments, mark the paths of the pool that do not include the request nodes with a *tabu tag* randomly selected in the interval [1, *tab_int*];
- If one or the other of the paths currently in the tour that correspond to the boarding and alighting nodes of the request do not include the node of the request: Swap the path with a non-tabu path from the pool and mark the existing path with a tabu tag randomly selected in the interval [1, *tab_int*] (it is a "normal" swap: one selects the time-feasible swap with the best score);
- Update Z_{best} if necessary; Goto 1

Otherwise, if $div > Max_div$ **Stop;** Otherwise, perform a **Diversification** phase:

- $div = div + 1; i = 0$
- Swap out of the best global tour *N_div* paths with the highest frequency in $\mathcal{F}^{\mathcal{G}}(p_h)$; Drop orphan requests and profitless nodes.
- Mark exiting paths with a *tabu tag* randomly selected in the interval [1, *tab_div*].
- Mend the tour.
- Goto 1.

Several variants of this basic framework have been defined and tested. The variants differ according to the particular *score* used to evaluate moves, *mending* procedure used to re-insert requests and node in the new tour, and strategy used to *select* the new tour between q_s and q_m, the solutions yielded by the swap and mending operations, respectively. We examine each of these algorithmic elements in turn.

5.1 Score functions

To compute the score $s(p_h^k)$ of a potential move, all orphan requests are dropped and, for the corresponding segment, the sum of the differences in the profits associated to the nodes of the paths and the associated transportation costs is calculated to obtain the *Exact* move evaluation $\Delta_{obj}(p_h^k)$. This is our basic score function. An alternative is given by the *Node potential* evaluation that implements the concept of potential node revenue introduced in Section 4. Note that the static and dynamic node potential benefit measures defined for the constructive heuristics, equations (17) and (18), yield the same result in a tabu search context. The node potential evaluation of a move is then computed as the sum of the exact evaluation and the potential revenue of the not yet satisfied requests associated to the nodes of the incoming path:

$$s(p_h^k) = \Delta_{obj}(p_h^k) + \sum_{i \in N(p_h^k)} \left(\sum_{r: i=s(r) \wedge d(r) \notin N(q^k)} u(r) \cdot \frac{\mathcal{S}^{hd}(r)}{\mathcal{S}^{pr}(r)} + \sum_{r: i=d(r) \wedge s(r) \notin N(q^k)} u(r) \cdot \frac{\mathcal{D}^{hd}(r)}{\mathcal{D}^{pr}(r)} \right). \tag{25}$$

where $N(p_h^k)$ and $N(q^k)$ stand for the sets of nodes of the candidate path and of the current tour excluding path h, respectively. The computation of the *promising* and *held* quantities for each request and node is carried out during the mending procedure presented next.

5.2 Mending and path improving procedures

Once paths are swapped and orphan requests and profitless nodes are dropped, the *mending* step attempts to insert nodes into an existing tour to increase the number of requests served and, therefore, the total benefit of the tour. It is a greedy hill climbing procedure performed as follows:

1. Compute the *request revenue* for all nodes not currently in the tour or not already examined and discarded;
2. Try to insert the node with the highest request revenue. If unsuccessful, try the second highest, then the third highest, and so on.
3. If successful, repeat the previous two steps until all nodes have been examined and either included in the tour or discarded.

Two ways of computing the request revenue have been defined. The *Absolute* variant computes the sum of the revenue of each request associated to it. The *Potential* variant sums up the *potential revenue*, $dpr(i)$ (equation 18), of the nodes. Costs are not considered in either version. Each time a node is selected for insertion, the $\mathcal{S}^{pr}(r)$ and $dnpr(r)$ memory values of all the requests associated to this node are increased by 1. The corresponding $\mathcal{S}^{hd}(r)$ and $\mathcal{D}^{hd}(r)$ values are updated only when the request is actually served by the new tour (i.e., the two nodes are inserted).

The mending is completed by a re-ordering of the tour. These operations create a potentially new path for each segment, which is added to the pool P'. The usual verifications and, possibly, updates of the best solution values are also performed.

5.3 Selection strategy and tabu search variants

Following a mending operation, two candidate solutions are available: q_s^k, the tour yielded by the swap move and the dropping of the orphan requests and profitless nodes, and the mended tour q_m^k. Two strategies have been considered to select the new current solution between these two candidates:

KS Keep q_s^k. This corresponds to the direct approach of continuing working from the result of the move performed in the neighbourhood defined initially, the mending operation being mainly used to enrich the pool.

KB Keep the tour with the best (higher) solution value. This appears as the natural approach with respect to the optimization objective. It does imply, however, that the tabu search structure presented before is slightly modified and the precise nature of the move finally implemented (and the neighbourhood finally explored) is known only the mending operation has been performed.

Variant	Basic	BB	PPKS	PPKB
Score function	Exact	Exact	Node potential	Node potential
Mending criterion	Absolute	Absolute	Potential	Potential
Selection Strategy	KS	KB	KS	KB

Table 1: The Tabu search Variants

Four variants of the tabu search framework have been defined based on these possibilities and are defined in Table 1. All variants start the search by performing the same **Initialization** procedure: n iterations of the basic tabu search - exact score evaluation, absolute node-request profit for mending, and KS selection strategy - with an initial pool that contains only the paths generated during the Lagrangian Relaxation procedure.

6 Hybrid Search Approaches

The meta-heuristic methodologies described in the two preceding sections yield constructive and improving procedures. It is only natural to envision methods that combine the two approaches and, hopefully, inherit their best characteristics.

Three hybrid structures have been defined in the present study:

1. **Consecutive**. It corresponds to a simple juxtaposition of the two procedures with an equal share of the total computation effort. The procedure starts with the multi-trial constructive heuristic, which is stopped after half the specified number of iterations or computing time. The tabu search procedure starts from the best solution identified during the constructive phase and continues for the other half of the total computing effort. The idea behind this basic approach is to stop the somewhat random search of the constructive search and attempt to improve from the best solution found.

2. **G_TS_Postopt**. This procedure implements the multi-trial heuristic described in Section 4, but the solution found at each iteration is improved by using the tabu search meta-heuristic. The tabu search thus replaces the post-optimization procedure described earlier on. For efficiency considerations, the tabu search is run for only one sequence of Neighbourhood Search and Intensification phases.

3. **TS_Gdiv**. The Tabu Search procedure is run with a different diversification scheme: one iteration of the construction procedure. The idea behind this approach is that diversifying the search is often a critical success factor is meta-heuristic search. The strategy does not correspond to a random re-start procedure since all long-term memories are kept and used.

In the three approaches, the pool constitutes a common resource, shared and enriched by each algorithm. Similarly, all memories are updated and kept from phase to phase to ensure the long term learning capabilities of the composite procedure. Each structure may yield four hybrids according to the particular tabu search variant used. Preliminary testing restricts this number as described in Section 7.

The procedures are designed to require similar computational efforts to ensure meaningful comparisons. Several objectives are in sight other than that of searching for the best solution method. We are interested, in particular, to evaluate the impact on the performance of the tabu search procedure of significantly improving the quality of the initial solution. We also aim to investigate the role of the constructive, multi-trial procedure, as main diversification move in memory-based meta-heuristics.

7 Computational Results

The goals of the experiments reported in this section were to (1) calibrate each meta-heuristic and select the best strategy (variant) for each method; (2) perform a meaningful comparison between the two meta-heuristics; (3) explore avenues of combining the two; (4) determine the best solution approach for the DAS1 problem in terms of solution quality and computing efficiency to provide good algorithmic building blocks for more general formulations.

Most calibration and comparison efforts have been performed on sets of problems derived from actual transit lines. The lines have been selected from the transit networks of one Canadian (identified as W) and one European (identified as S) city, together with representations of the corresponding street networks. Each line has been the seed for several problem instances through variations in the number of segments, optional stops for each segment, the density of requests, the width of time windows, and the time available to travel each segment. The actual length of each street represented (through a link) in each of the two networks is known. The triangle inequality relations hold for the street lengths, but distances are neither Euclidean nor symmetric. This information has been used to compute distances between all the nodes in the network by way of an all-pairs shortest path algorithm.

For each transit line, a subset of stops (one stop for each segment) has been extracted from the original definition to act as the compulsory stop set of the flexible line. Each street intersection laying within a maximum distance from any stop of the original transit line is eligible as an optional stop of the flexible line. Optional stops are then assigned to the respective segments. Costs and times on the arcs of the network have been defined proportional to distances. This not only reflects rather widely accepted practice, it also allows the algorithm to discriminate among paths serving the same node set in different sequences and to privilege the most efficient ones with respect to time. The benefits associated to requests have been defined such that the extra cost of any detour to serve a customer is always compensated by the additional benefit.

With such settings, limited time availability is the only constraint that may cause requests to be discarded. Time windows at compulsory stops have been set in such a way that extra time is provided for detours within each segment. In fact, for each segment h, the maximum duration of a path $(b_{h+1} - a_h)$ has been set equal to one and a half times the duration of the path from compulsory stop f_h to f_{h+1} in the original transit line. Time window width at compulsory stops varies from 60 to 120 seconds, which is considered a reasonable waiting time for passive users who do not know in advance the arrival time of the vehicle. The number of requests is proportional to the number of segments (25 requests for each segment), and boarding and alighting stops have been randomly chosen in the node set N according to a uniform distribution. This decision is intended to reproduce the most difficult situation, where there are no substantial differences in the number of requests associated to each optional node.

Some basic considerations hold. Instances with more segments are more difficult. Indeed, due to the time relations linking adjacent segments, local decisions on path duration induce

chain reactions. Thus, the longer the chain, the larger the impacts. Other factors being equal, the difficulty is also related to the dimension of the geographical area encompassed by each segment. This attribute is controlled by the maximum distance parameter used to select the optional stops. Considering these factors, several classes of instances have been defined to depict scenarios of increasing difficulty. In particular, we generated problem instances with 5 and 10 segments and with two different values (0.3 and 0.4) for the maximum distance allowed for optional stops. The number of optional stops in each instance varies accordingly, increasing with the number of segments and with the maximum distance allowed.

Different sets of problems have been generated for the calibration phase (19 problems) and for the actual experimentation (43 problem instances). The transit lines in the European network are very long while the associated graph is rather sparse (measured in the number of nodes). The opposite is true for the Canadian case. Thus, given the 25 requests/segment condition, the 23 W problems of the final data set have all 10 segments, while the 20 S problem instances have 5 segments each. The complete test problem data may be obtained from the authors. In order not to over charge the paper, in the following only aggregated figures are presented.

The goal of the calibration phase was to determine for each of the two meta-heuristics parameter values that offer a consistently good average behaviour over the set of problems tested. When several sets of parameters displayed similar behaviours, the selection was biased towards those that displayed superior performances when best results were considered.

Different variants of the constructive procedure were obtained by switching on and off the bonus, perturbation, and memory probability smoothing factors. The experimentation has shown that all $\rho(p)$, $\rho(i)$, and α parameters are required for good performance of the constructive heuristic. In particular, $\beta = 0.5$ and $\alpha = 0.98$.

The following parameter values have been considered in the calibration of the tabu search methods:

- Length of the short-term tabu list: [10,20] and [20,30];
- Number of consecutive unimproving moves before initiating an Intensification phase: 20, 30, 40, and 50;
- Intensification phase: included (1) or not (0);
- Number of paths swapped to initiate a Diversification phase (N_div): 1, 3, and 4;
- Maximum number of iterations: 500, 800, 1000.

As a general conclusion of the calibration phase, it appears that longer searches yield better results. The gain is not very large though, and good results are obtained quite early in the search. It was also interesting to note that several parameter settings yield similar results, which emphasizes the robustness of the method with respect to parameter calibration.

The selected parameter settings for each tabu search variant are (presented in the order of the previous paragraph):

- **Basic**: [20, 30], 50, 1, 4, 1000
- **BB**: [20, 30], 30, 1, 4, 1000
- **PPKS**: [20, 30], 20, 1, 4, 1000
- **PPKB**: [20, 30], 40, 1, 4, 1000

Hybrids have not been calibrated. Instead, the parameter settings determined for each meta-heuristic were also used for the hybrid procedures. The calibration phase was used, however, to reduce the number of procedures admitted to the experimentation phase. We analyzed the comparative behaviour of the constructive heuristic, the four tabu search versions, and the three hybrid strategies, all three using the basic tabu search method. All methods were run for 1000 iterations. The results tend to indicate that the **Basic** and **PPKS** tabu search versions perform better than the two others. An explanation of this apparently inconsistent behaviour (the **BB** and **PPKB** methods always keep the best tour after mending) resides in the fact that the selection of the best tour after mending corresponds to a move to a local best solution. And it may take several iterations to move out of the region of a local optimum and find another improving solution. Variants **BB** and **PPKB** were thus discarded from the following experimentation rounds.

The three hybrid approaches offered very good performances but displayed different behaviours. In general, the third appeared to outperform the first two. For the first one, the flaw appears to reside in the fact that it allows less exploration time to both meta-heuristics. Indeed, we observed that when longer exploration times were permitted (e.g., 60 seconds), the first hybrid strategy tended to perform very well, competitively so with respect to the third strategy. As for the second hybrid, although improving by tabu search appears superior to a simple post-optimization phase, the global search does not seem as efficient as when based on a tabu search framework, as is the case for the third hybrid. Thus, only hybrids built on the pattern of the third one were considered in the next experimental phase.

The five meta-heuristics that have been included in the experimental phase therefore were the constructive meta-heuristic (identified as **MMG** in the following), the **Basic** and **PPKS** versions of the tabu search, as well as two versions of the third hybrid, **TS_Gdiv** and **PPKS_Gdiv**, using the **basic** and **PPKS** versions of the tabu search, respectively.

All procedures include a number of random parameters. Consequently, during the calibration phase, each experiment was repeated five times for each problem. The number of repetitions has been increased to ten for the actual experimentation. The procedures have proved very robust, however, Thus, for the vast majority of problem instances solved with the simple meta-heuristics **MMG** and **Basic**, the standard deviation computed over the ten repetitions represented less than 1% of the corresponding average,the maximum being around 2%. For the **PPKS** procedure the figures are 2% and 3%, respectively, while they

increase to 3% and 4% for the two hybrids. In all cases, variations are not meaningful and the average value is a good indication of the behaviour of the corresponding procedure. Only average values are thus considered in the following.

We are interested in good solutions in short resolution times, since the methods examined in this paper are to form one of the basic algorithmic blocks for more comprehensive DAS cases. On the other hand, longer computation times allow a better assessment of the rate of solution improvement of each method. Consequently, solution values have been collected after 1, 5, and 10 to 100 by increments of 10 CPU seconds. The meta-heuristics are coded in C and have been run on a PC Pentium II 233 MHz. Detailed results are included in the two appendices. In the following, we present aggregated statistics and performance results for the best procedures only.

The first issue is to compare the performance of the five meta-heuristics. Appendix 1 displays for each problem instance the solution value of each meta-heuristic at each of the twelve CPU time instances. Results are given both in table and graphic form. To synthesize this information, for each problem instance and solution time, we assigned a value of 5 to the best solution, of 4 to the second best, and so on until the lower solution received a value of 1. We then computed a global score for each method by adding up the values corresponding to its solutions for all problems. The resulting figures are displayed in Table 2 that indicates, for each method and CPU time limit, the global score, as well as the number of first, second, and third places. The corresponding evolution of the global scores with respect to the CPU times is illustrated in Figure 4.

The superior performance of the two hybrids and the **PPKS** tabu search variant appears clearly. One may be surprised, however, to notice that, apparently, the **PPKS** tabu search outperforms the two hybrids. A closer analysis taking into account the characteristics of the problem instances clarifies this point. Table 3, illustrated in Figures 5 and 6, separate the previous information between the two sets of problem instances.

Problems in set S correspond to very long lines (most being over 20 km long) and the associated network is rather sparse. This characteristic is emphasized by the fact that, in order to achieve the average of 25 optional stops per segment, only five segments per line could be defined for problems in this set. Consequently, the number of alternate routings is rather low and the problem becomes "easy". In this case, as illustrated in Table 3 and Figure 6, the tabu search that makes use of the request satisfaction long term memories and the associated node potential revenue performs best. The performance of the two hybrids is comparable but significantly lower than that of the **PPKS** tabu search. The fact that the **PPKS** method pacts more iterations in the same computing time is a contributing factor to its superiority in this context. The simple methods do not perform well, the **basic** tabu search failing miserably.

Problems in set W display characteristics closer to those of "normal" transit lines in urban environments. The length of most lines is less than 10 km (almost all are less than 15 km long) and a fairly dense street network. Most lines thus display ten segments with some 25

CPU time	MMG	Basic	PPKS	TS_Gdiv	PPKS_Gdiv
1	111	116	180	129	122
	8,6,3	10,6,5	23,9,8	0,14,16	3,12,10
5	101	104	166	152	139
	6,7,2	7,5,4	20,7,7	6,16,13	5,13,14
10	96	104	166	153	142
	3,9,4	8,4,3	21,5,7	5,20,13	7,11,15
20	94	100	172	149	146
	3,8,5	8,3,1	23,5,8	5,16,16	5,17,12
30	92	102	170	153	143
	4,5,6	7,4,3	22,6,7	7,15,16	4,17,12
40	88	95	169	150	144
	4,5,4	5,3,6	22,6,9	7,16,13	5,18,9
50	85	93	168	145	137
	4,3,7	6,3,4	21,7,9	7,15,12	5,16,9
60	89	98	170	158	146
	4,3,7	6,3,5	22,6,7	8,18,12	5,17,12
70	89	99	169	156	147
	4,4,6	5,5,4	21,6,9	7,19,11	7,14,13
80	88	98	169	155	149
	4,3,5	5,2,5	20,2,10	7,11,12	8,9,12
90	88	99	171	156	147
	4,3,5	5,2,6	21,2,9	7,11,11	7,10,11
100	87	99	171	158	147
	4,4,4	5,4,6	21,7,9	7,20,11	7,15,11

Table 2: Global Score

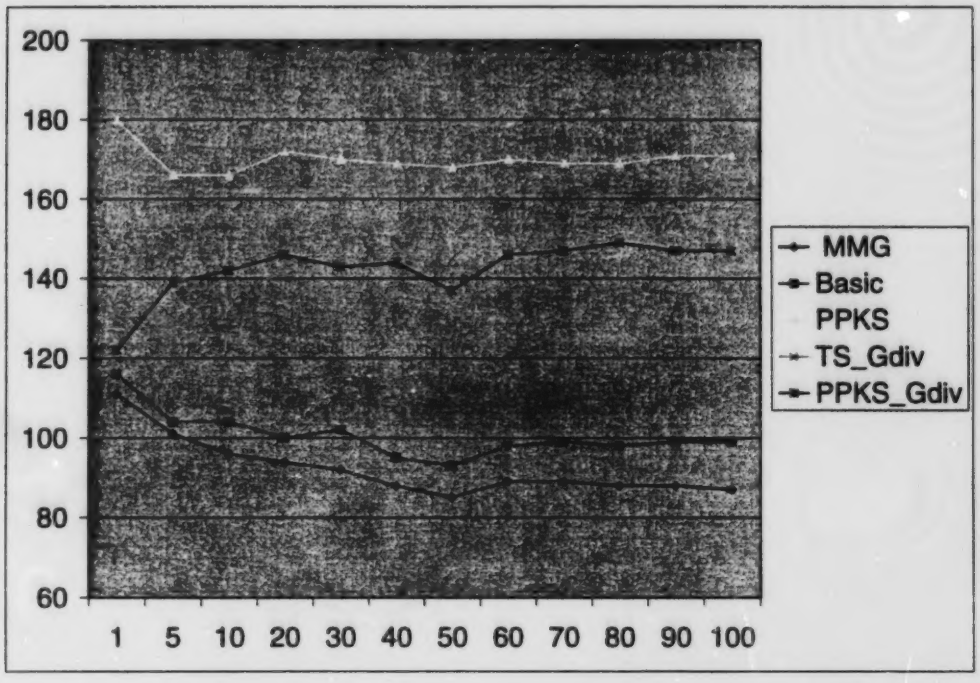


Figure 4: Evolution of Global Scores with CPU Times

CPU time	Problems W					Problems S				
	MMG	Basic	PPKS	TS Gdiv	PPKS Gdiv	MMG	Basic	PPKS	TS Gdiv	PPKS Gdiv
1	64	84	87	66	56	47	32	93	63	66
1	5,4,3	10,5,2	8,6,6	0,7,7	1,4,5	3,2,0	0,1,3	15,3,2	0,7,9	2,8,5
5	47	75	76	89	72	54	29	90	63	67
5	2,3,1	7,4,4	7,3,5	6,10,6	2,8,6	4,4,1	0,1,0	13,4,3	0,8,7	3,5,8
10	43	75	77	87	76	53	29	89	66	66
10	0,5,1	8,3,3	8,1,5	5,10,7	3,8,7	3,4,3	0,1,0	13,4,2	0,10,6	4,3,8
20	41	71	76	88	82	53	29	96	61	64
20	0,4,2	8,2,1	7,1,8	5,10,7	4,10,5	3,4,3	0,1,0	16,4,0	0,6,9	1,7,7
30	40	71	75	91	81	52	31	95	62	62
30	0,3,3	7,3,2	7,1,7	7,8,8	3,12,3	4,2,3	0,1,1	15,5,0	0,7,8	1,5,9
40	37	64	74	86	82	51	31	95	64	62
40	0,3,1	5,2,5	7,1,9	7,8,5	4,12,3	4,2,2	0,1,1	15,5,0	0,8,8	1,6,7
50	37	64	78	84	79	48	29	90	61	58
50	0,3,2	6,2,3	7,2,9	7,7,5	4,11,2	4,0,5	0,1,1	14,5,0	0,8,7	1,5,7
60	40	66	75	93	86	49	32	95	65	60
60	0,3,2	6,2,4	7,1,7	8,9,5	4,12,4	4,0,5	0,1,1	15,5,0	0,9,7	1,5,
70	39	66	74	93	87	50	33	95	63	60
70	0,3,2	5,3,4	6,1,9	7,12,2	6,9,5	4,1,4	0,2,0	15,5,0	0,7,9	1,5,8
80	38	64	74	93	89	50	34	95	62	60
80	0,3,1	5,2,4	5,2,10	7,11,4	7,9,4	4,1,1	0,2,1	15,5,0	0,7,8	1,5,8
90	38	64	76	93	88	50	35	95	63	59
90	0,3,1	5,2,4	6,9,9	7,11,4	6,10,4	4,1,4	0,2,2	15,5,0	0,8,7	1,5,7
100	38	64	76	94	88	49	35	95	64	59
100	0,3,1	5,2,4	6,2,9	7,12,3	6,10,4	4,1,3	0,2,2	15,5,0	0,8,8	1,5,7

Table 3: Global Score by Problem Class: "Normal" (W) and "Easy" (S)

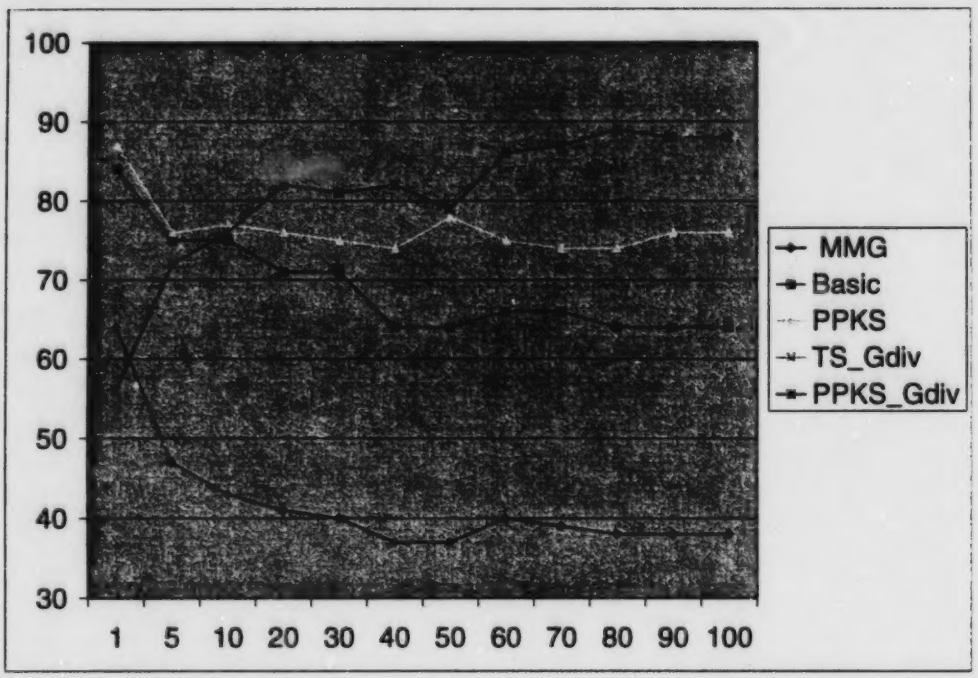


Figure 5: Evolution of Global Scores with CPU Times - "Normal" Problems (Set W)

optional stops per segment. The number of alternate paths increases as does the differences among request in terms of boarding alighting optional stops. This and the higher number of segments makes for significantly harder problem instances. The hybrids, see Table 3 and Figure 5, perform best in this context. The combination of aggressive improvement (tabu search) and diversification (memory-based construction) pays off. It is interesting to note that the **TS_Gdiv** hybrid slightly out performs the **PPKS_Gdiv** method. Both methods make use of the same diversification mechanism based on the long term node potential memories. It would then appear that, in this context, the most profitable approach is to make a simpler but more thorough tabu search exploration and leave the integration of long term request-satisfaction statistics to the diversification mechanism. It is also interesting to note that the PPKS tabu search performs rather well, although not on par with the hybrids. The simple tabu search is rapidly outclassed, while the memory-based construction heuristic cannot compete.

These observations hold over the entire range of CPU times considered. Our conclusion is therefore to use the **PPKS** tabu search method for the "easy" problems and the **TS_Gdiv**

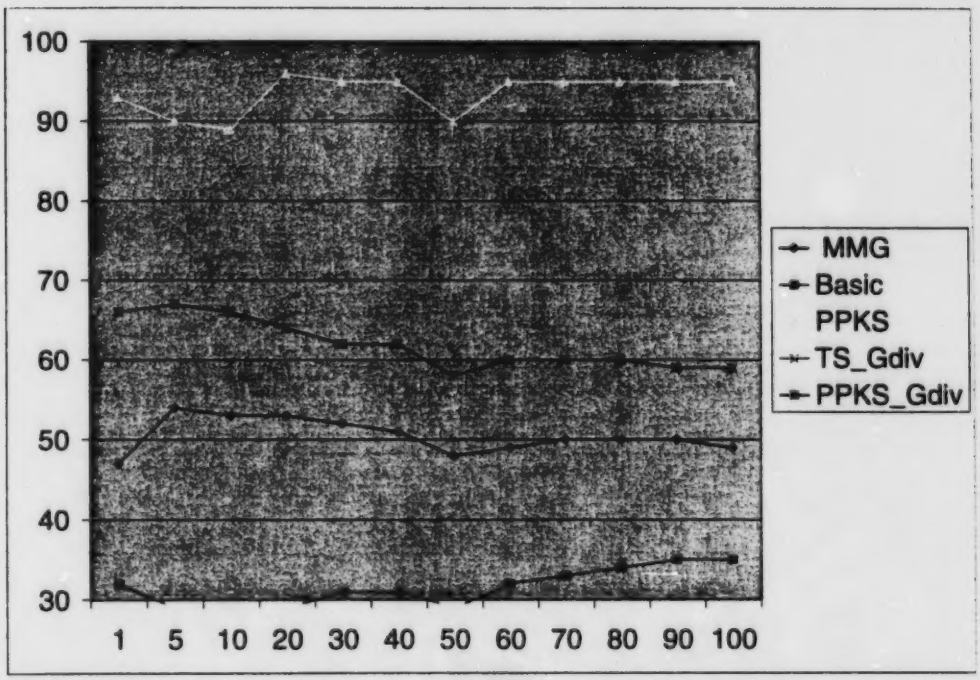


Figure 6: Evolution of Global Scores with CPU Times - "Easy" Problems (Set S)

hybrid for all other cases. The question of the performance of the methods relative to the optimal solution still stands, however.

To analyze the quality of the meta-heuristics, we examined the possibility to determine the optimal values of the test problems or good quality upper bounds (the Lagrangian upper bound offers an indication, but not necessarily a very good one). To solve directly the path-based network design problem (2) - (10) requires algorithmic developments that go beyond the needs of the project. We thus wrote an arc-based formulation [19] that may be approached by using standard software tools. For us, it was CPLEX 6.6 run for up to for 10 hours CPU time on 10 of the 64-processors of a SUN E10000 computer (with 64 Gb of RAM memory and 400 Mhz processors running under Solaris 2.7). The branch-and-bound procedure of CPLEX has been configured to generate all possible cuts and to aim for the best bound possible (*best first* option).

Appendix 2 gives, in table and graphic formats, the gaps in percentage of each solution value relative to the best upper bound identified by CPLEX after 100 hours of CPU time. Table 4 synthesizes these results. Columns **PPKS 10** and **TS_Gdiv 10** give the relative gaps obtained by the two methods after 10 seconds of CPU time. Columns **PPKS 100** and **TS_Gdiv 100** display the same information after 100 seconds of CPU time. Finally, column **B&B** displays the gap between the upper and lower bounds computed by CPLEX. For the problem instances for which the branch-and-bound has found the optimal solution, column **B&B** gives the corresponding total CPU time (in seconds). The *inf* for problem W18 signals that no feasible solution has been found by the CPLEX B&B within the time allocated. Several conclusions emerge.

It is noteworthy that the experiments illustrate the complexity and computational difficulty of the class of formulations generated by DAS1 problems. For many problem instances, the gap between the upper and lower bound computed by CPLEX is still very high after a significant amount of computational effort. Only for easy problems is the optimal solution found after a significant computational effort. (Note that a few W problems are "easy" as well due to a lower number of optional stops per segment.) And no feasible (integer) solution has been found for one problem instance (W17).

The meta-heuristics perform very well in these circumstances. They offer consistently very high quality solutions with very low computational efforts. In fact solutions are already very good after 10 seconds of CPU time, and are almost as good after only 5 seconds. The methods are not always in a "steady" behaviour after 5 seconds, however, Thus, we will probably use 10 seconds when addressing general DAS formulations. In a tiny fraction of the branch-and-bound time, the meta-heuristics "solve" the easier problems to near optimality and find excellent solution to the hardest ones we addressed.

Prob	PPKS 10	TS_Gdiv 10	PPKS 100	TS_Gdiv 100	B&B
W1	0.04	0.13	0.03	0.13	(145.42s)
W2	10.66	10.72	9.98	9.98	35.88
W3	2.46	2.97	1.78	1.16	2.31
W4	4.69	4.21	4.07	3.76	39.16
W5	5.80	4.70	4.12	3.46	11.07
W6	8.37	7.89	7.48	7.10	30.39
W7	5.56	6.54	5.17	5.25	34.06
W8	9.45	10.04	8.67	8.57	31.88
W9	4.34	5.18	3.94	3.97	15.16
W10	0.00	0.00	0.00	0.00	(0.14s)
W11	0.08	0.28	0.02	0.28	(0.05s)
W12	4.56	4.50	3.24	3.05	15.01
W13	5.75	4.77	4.69	4.43	23.70
W14	4.05	3.95	2.82	2.54	3.60
W15	3.99	3.87	3.43	3.19	22.64
W16	14.11	13.17	13.43	12.31	42.33
W17	9.57	8.20	8.92	7.96	24.27
W18	11.07	11.83	11.04	11.13	inf
W19	11.05	10.35	9.61	8.63	29.95
W20	5.29	5.09	4.95	4.97	35.57
W21	10.54	9.70	10.09	9.14	30.64
W22	1.95	1.51	1.88	1.13	(5950.55s)
W23	0.06	0.18	0.05	0.16	(13911.92s)
S1	0.88	1.77	0.75	1.67	1.59
S2	4.16	4.45	4.05	4.30	30.99
S3	0.40	1.01	0.35	0.79	(787.61s)
S4	0.27	0.60	0.12	0.52	(107222.16s)
S5	0.15	0.45	0.09	0.42	(52802.27s)
S6	2.20	2.49	2.20	2.49	9.37
S7	2.91	3.16	2.75	3.10	27.87
S8	6.86	7.27	6.39	6.77	(8834.61s)
S9	0.39	0.55	0.11	0.31	(27484.77s)
S10	2.04	1.94	1.66	1.78	1.88
S11	0.28	0.70	0.22	0.65	(3947.70s)
S12	0.40	1.48	0.10	1.21	(8344.85s)
S13	12.15	14.05	12.13	13.52	2.33
S14	0.06	1.14	0.00	0.95	(2643.30s)
S15	1.10	2.22	1.10	2.11	8.92
S16	4.29	4.64	3.95	4.21	27.93
S17	0.91	1.44	0.80	1.22	13.30
S18	0.05	0.63	0.03	0.24	(4795.73s)
S19	0.40	0.52	0.01	0.42	(2667.44s)
S20	8.64	8.86	8.54	8.54	58.72

Table 4: Comparative Performances Gaps

8 Perspectives and Conclusions

In this paper two different families of meta-heuristic strategies have been proposed for the solution of the basic version of a new flexible transport system: a memory-enhanced, multi-trial, randomized constructive procedure and a tabu search method. Several variants of these strategies have been developed and analyzed. Moreover, several hybrid strategies combining the two meta-heuristic strategies have also been developed and analyzed. Particular care has been paid to ensure a best-implementation effort for all procedures. Thus, at our knowledge at least, this is one of the rare systematic and fair comparative studies of these meta-heuristics.

Both basic methodologies allow to determine good solutions with limited computation efforts. Enhancing the randomized multi-trial heuristic with long-term memories appears quite beneficial, essential even. Tabu search appears to dominate the multi-trial constructive heuristic, however. Enhancing the tabu search procedure with additional memories and probabilistic evaluations of potential benefits significantly enhances its performance.

The constructive heuristic, and the increased randomization it brings to the search, appears to offer a superior diversification strategy. The resulting hybrid search thus emerges as the method of choice.

The resolution of DAS1 is only a first step in addressing the planning issues related to general demand-responsive, flexible transit systems. We are now working on various settings of DAS, where the efficient resolution of DAS1 is a pre-requisite. We hope to report on these efforts shortly.

Acknowledgments

We express our gratitude towards Professor Michael Florian who graciously gave us access to a number of data banks from which a set of realistic test problems could be derived. We also thank Marc-André Morissette and Riad Benghanen who helped out with the long and tedious experimental phase.

This project is supported by grants from NSERC Canada, the Fonds FCAR of the Province of Québec, the Québec-Italy cooperation program of the Ministry for Foreign Affairs of Italy and the Ministry for International Relations of the Province of Québec, and the *Progetto strategico: Metodologie per la logistica* of the CNR Italia.

We also acknowledge the combined support of the Canada Foundation for Innovation, the government of the Province of Québec, SUN Microsystems, Inc. and NSERC Canada in providing the computation facilities of the Centre for Research on Transportation that we used.

References

[1] Borndörfef, R., Gröetschel, M., Klostermeier, F., Küttner, C., *Telebus Berlin: Vehicle Scheduling in a Dial-a-Ride System*, Konrad-Zuse Center for Information Systems SC-97-23, 1997.

[2] Carraresi, P., Frangioni, A., Nonato, M., *Applying Bundle Methods to Optimization of Polyhedral Functions: An Applications Oriented Development*, **Ricerca Operativa** 25, 5-49, 1996.

[3] Crainic T.G., Malucelli F., Nonato M., *Flexible Many-to-few + Few-to-namy = an Almost Personalized Transit System*, **Preprints TRISTAN IV - Triennial Symposium on Transportation Analysis**, faculdade de Ciências da Universidade de Lisboa and Universidade dos Açores, June 13-19, ão Miguel, Açores, Portugal.

[4] Crainic, T.G., Gendreau, M., Soriano, P., Toulouse, M., *A Tabu Search Procedure for Multicommodity Location/Allocation with Balancing Requirements*, **Annals of Operations Research** 41, 359-383, 1993.

[5] Crainic, T.G., Gendreau, M., Farvolden, J.M., *A Simplex-Based Tabu Search Method for Capacitated Network Design*, **INFORMS Journal on Computing** 12(3), 27-46, 2000.

[6] Crainic T.G., Guertin F., Malucelli F., Nonato M., *Adaptive Memory Programming for a Class of Demand-Responsive Transit Systems*, **Proceedings CASPT 2000 - 8th International Conference on Computed Aided Scheduling of Public Transport**, Berlin, Germany, June 21-23, 2000, to appear.

[7] Crainic T.G., Malucelli F., Nonato M., *A Demand Responsive Feeder Bus System*, CD-ROM of **7th World Congress on Intelligent Transport Systems**, Torino, Italia.

[8] Desrosiers, J., Dumas, Y., Solomon, M.M., Soumis, F., *Time Constrained Routing and Scheduling*, In *Network Routing, Handbooks in Operations Research and Management Science* 8, M.O. Ball, T.L. Magnanti, C.L. Monma, and G.L. Nemhauser (Eds.), 35-139, 1995, North-Holland, Amsterdam.

[9] Ioachim, I, Desrosiers, J., Dumas, Y., Solomon, M.M., Villeneuve, D., *A Request Clustering for Door to Door Handicapped Transportation*, **Transportation Science** 29, (1995) 63–78.

[10] Feo, A.T., Resende, M.G.C., Smith, H.S., *A Greedy Randomized Adaptive Search Procedure for Maximum Independent Set*, **Operations Research** 42(5), 860-878, 1994.

[11] Feo, T.A., Resende, M.G.C., *Greedy Randomized Adaptive Search Procedures*, **Journal of Global Optimization** 6(2), 109-133, 1995.

[12] Gendreau, M., Laporte, G., and Potvin, J.-Y., *Metaheuristics for the Vehicle Routing Problem* in *The Vehicle Routing Problem*, P. Toth and D. Vigo (Eds.), SIAM, 2000, forthcoming.

[13] Glover, F., *Tabu Search - Part I*, **ORSA Journal on Computing** 1(3), 190-206, 1989.

[14] Glover, F., *Tabu Search - Part II*, **ORSA Journal on Computing** 2(1), 4-32, 1990.

[15] Glover, F., Laguna, M., *Tabu Search*, Kluwer, Norwell, MA, 1997.

[16] Laporte, G., *The Traveling Salesman Problem: An Overview of Exact and Approximate Algorithms*, **European Journal of Operational Research** 59, 231-247, 1992.

[17] Laporte, G., *The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms*, **European Journal of Operational Research** 59, 345-358, 1992.

[18] Lawler, E.L., Lenstra, J.K., Rinnoy Kan, A.H.G., Shmoys, D.B., *The Traveling Salesman Problem: A Guided Tour of Combinatorial Optimization*, John Wiley, New York, N.Y., 1985.

[19] Malucelli, F., Nonato, M., *Formulations and Bounding Procedures for Optimization Problems Arising in Managing Demand Adaptive Transit Systems*, Working paper, Politecnico di Milano, 2001.

[20] Malucelli, F., Nonato, M., Pallottino, S., *Some proposals on flexible transit*, in T.Ciriani et al.(eds.), *Operations Research in Industry*, McMillian (1999).

[21] Salvensbergh, M.W.P., Sol, M., *The General Pickup and Delivery Problem*, **Transportation Science** 29(1), 17-29, 1995.

[22] Shen,Y., Potvin, J.-Y., Rousseau, J.-M., Roy, S., *A Computer Assistant for Vehicle Dispatching with Learning Capabilities*, **Annals of Operations Research** 61, 189–211, 1995.

Appendix 1

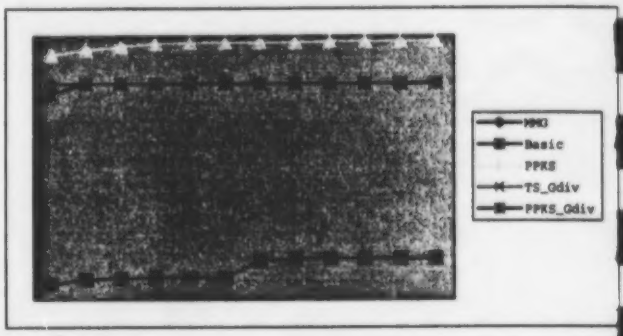
This appendix includes the detailed experimental results for the five selected meta-heuristics: the memory-enhanced, randomized greedy constructive procedure MMG, the **Basic** tabu search, the tabu search approach that makes use of long-term request-satisfaction memories and probabilistic node potential revenue **PPKS**, as well as the two hybrids **TS_Gdiv** and **PPKS_Gdiv** that combine the respective tabu search approaches with the MMG method as a diversification device.

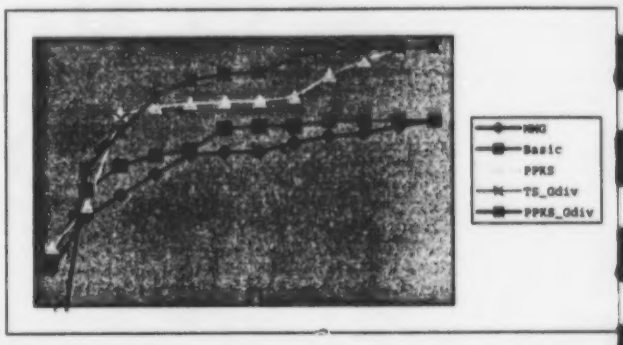
Results are given for twelve levels of computational efforts: 1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 seconds of CPU time on a Pentium II 233 Mhz PC.

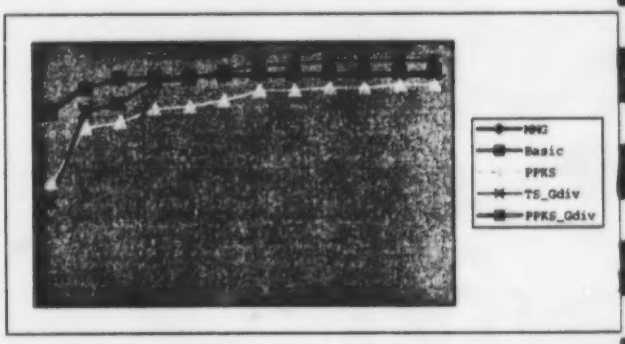
For each problem instance and each level of computational effort the solution value is indicated for each of the five meta-heuristics. The evolution of these solution values with the CPU time is also graphically displayed.

Finally, for each problem instance, the best upper bound identified by the branch-and-bound method of CPLEX 6.6 is also shown. A * identifies an optimal solution. The branch-and-bound algorithm has been run for 10 hours (wall-clock time) of computing time on 10 of the 64 processors of a SUN E10000 computer (with 64 Gb of RAM memory and 400 Mhz processors running under Solaris 2.7).

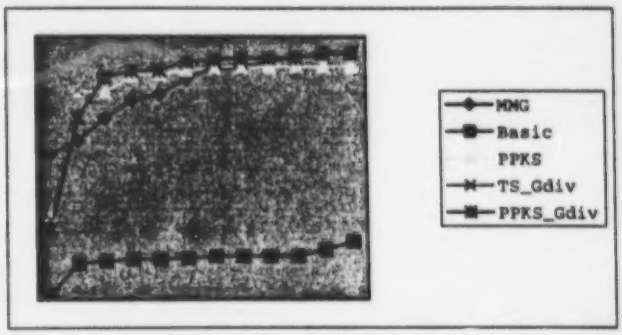
time	MMG Average	Basic Average	PPKS Average	TS_Gdiv Average	PPKS_Gdiv Average
W1	BB =	278,96	*		
1	278,61	277,39	278,78	278,21	278,55
5	278,61	277,416	278,82	278,61	278,61
10	278,61	277,425	278,84	278,61	278,61
20	278,61	277,43	278,86	278,61	278,61
30	278,61	277,43	278,86	278,61	278,61
40	278,61	277,43	278,86	278,61	278,61
50	278,61	277,53	278,86	278,61	278,61
60	278,61	277,545	278,86	278,61	278,61
70	278,61	277,545	278,87	278,61	278,61
80	278,61	277,545	278,87	278,61	278,61
90	278,61	277,545	278,87	278,61	278,61
100	278,61	277,545	278,87	278,61	278,61
W2	BB =	217,7			
1	189,94	190,219	190,52	188,60	186,33
5	191,32	192,011	191,60	192,44	192,57
10	191,88	192,655	194,02	193,90	193,54
20	192,44	192,928	194,20	194,74	194,45
30	192,94	193,053	194,30	194,86	194,91
40	193,00	193,616	194,32	195,06	195,04
50	193,01	193,675	194,35	195,10	195,10
60	193,24	193,677	194,45	195,13	195,47
70	193,42	193,809	195,02	195,52	195,70
80	193,49	193,814	195,41	195,53	195,74
90	193,67	193,819	195,69	195,69	195,74
100	193,80	193,819	195,69	195,69	195,74
W3	BB =	211,15			
1	206,51	206,322	203,68	199,16	202,95
5	207,33	207,257	205,75	203,39	206,38
10	207,58	207,666	205,96	204,87	206,69
20	207,59	207,695	206,47	207,00	207,50
30	207,71	207,721	206,57	208,47	207,61
40	207,71	207,782	206,79	208,60	207,92
50	207,72	207,801	207,16	208,60	208,14
60	207,72	207,801	207,18	208,60	208,17
70	207,72	207,801	207,26	208,62	208,27
80	207,72	207,808	207,27	208,70	208,29
90	207,72	207,819	207,38	208,70	208,29
100	207,72	207,827	207,38	208,70	208,29

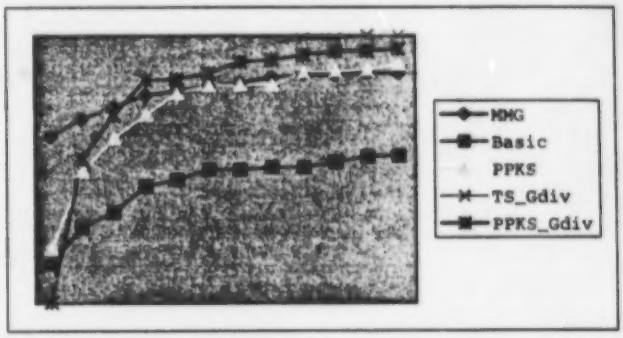


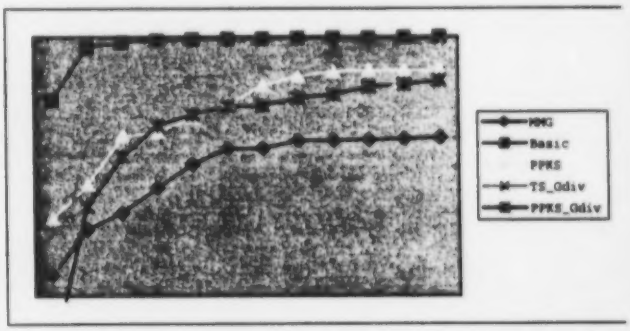




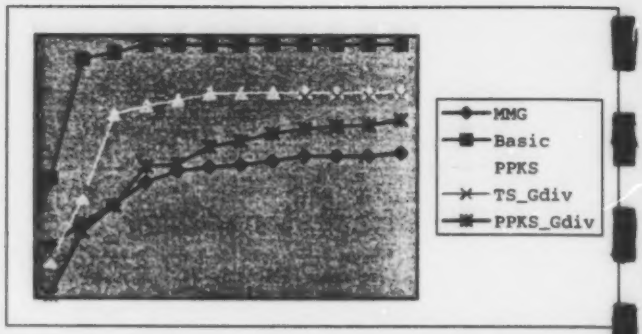
time	MMG Average	Basic Average	PPKS Average	TS_Gdiv Average	PPKS_Gdiv Average
W4	BB =	227,7			
1	213,70	206,286	210,29	210,19	209,71
5	214,45	207,893	217,14	217,19	215,68
10	215,62	208	217,03	218,12	217,80
20	216,62	208,127	217,92	218,45	218,16
30	216,94	208,128	217,98	218,60	218,20
40	217,74	208,15	218,18	218,63	218,67
50	218,31	208,277	218,35	218,87	218,78
60	218,31	208,277	218,39	218,93	218,86
70	218,67	208,277	218,41	218,94	218,86
80	218,88	208,284	218,41	218,94	218,86
90	219,18	208,612	218,41	219,13	219,10
100	219,18	209,049	218,42	219,13	219,10
W5	BB =	227,26			
1	214,17	207,537	208,30	206,50	205,52
5	215,09	209,438	212,35	212,82	212,98
10	215,71	210,197	214,07	216,58	215,25
20	216,30	211,575	215,29	217,86	217,10
30	216,53	211,898	216,43	218,12	217,13
40	216,81	212,411	216,83	218,49	217,46
50	216,95	212,421	216,84	218,76	218,03
60	217,20	212,525	216,84	219,05	218,13
70	217,35	212,534	217,50	219,21	218,38
80	217,35	212,809	217,55	219,22	218,61
90	217,44	213,11	217,61	219,39	218,62
100	217,45	213,116	217,89	219,39	218,71
W6	BB =	245,96			
1	220,69	226,588	222,59	220,28	218,11
5	222,23	228,34	223,74	225,14	222,99
10	222,78	228,532	225,37	226,55	224,67
20	223,63	228,592	225,49	226,62	225,73
30	224,41	228,616	225,98	227,35	226,09
40	224,94	228,637	226,44	227,43	226,36
50	224,94	228,637	227,08	227,57	226,38
60	225,21	228,659	227,38	227,66	226,63
70	225,21	228,659	227,54	228,31	226,75
80	225,23	228,662	227,57	228,31	227,01
90	225,26	228,682	227,57	228,48	227,09
100	225,30	228,704	227,57	228,49	227,23

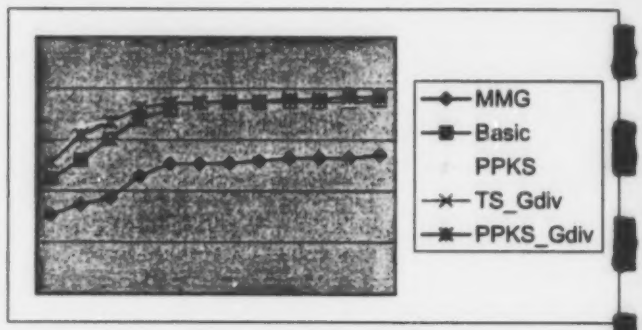


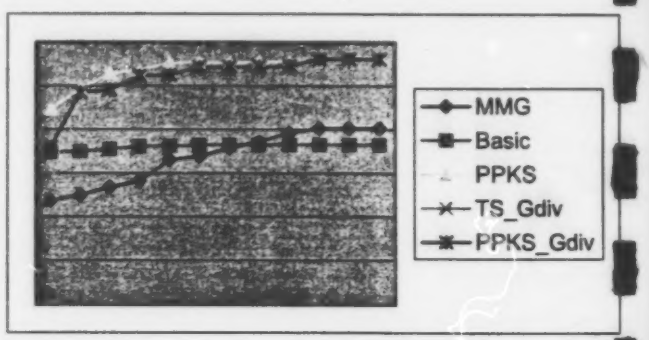




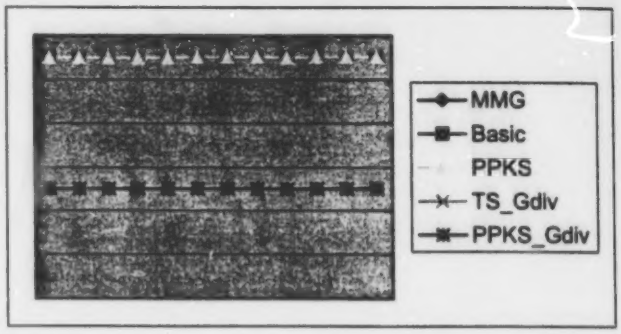
time	MMG Average	Basic Average	PPKS Average	TS_Gdiv Average	PPKS_Gdiv Average
W7	BB =	228,18			
1	210,51	213,057	209,88	210,09	208,62
5	211,31	217,601	212,30	212,53	211,00
10	212,05	217,831	215,50	213,25	212,02
20	212,90	218,127	215,83	214,08	213,52
30	213,33	218,135	216,04	214,71	213,59
40	213,49	218,135	216,29	215,03	214,25
50	213,57	218,143	216,32	215,62	214,48
60	213,73	218,157	216,34	215,69	214,78
70	213,91	218,157	216,34	216,16	214,93
80	213,91	218,157	216,34	216,17	215,05
90	213,92	218,157	216,34	216,17	215,09
100	214,03	218,157	216,38	216,19	215,29
W8	BB =	272,74			
1	237,71	241,188	242,66	241,19	242,77
5	238,68	242,986	246,31	244,49	245,41
10	239,31	245,091	246,95	245,37	246,75
20	241,48	246,983	248,29	247,39	248,14
30	242,66	247,922	248,57	248,07	248,55
40	242,67	248,63	248,79	248,39	248,64
50	242,72	248,66	248,80	248,65	248,76
60	242,96	248,66	248,84	248,83	248,76
70	243,19	248,707	248,88	249,06	248,93
80	243,19	248,707	248,88	249,09	248,94
90	243,19	248,737	248,89	249,38	249,14
100	243,57	248,85	249,09	249,38	249,14
W9	BB =	174,12			
1	160,73	162,965	164,92	162,19	163,16
5	161,00	162,999	165,88	164,87	165,68
10	161,35	163,091	166,57	165,11	165,83
20	161,64	163,229	166,72	166,16	166,45
30	162,61	163,253	167,06	166,21	166,49
40	162,76	163,267	167,07	166,64	166,85
50	163,14	163,289	167,08	166,64	166,85
60	163,43	163,302	167,10	166,64	166,89
70	163,84	163,302	167,13	167,13	166,89
80	164,06	163,302	167,14	167,13	167,21
90	164,06	163,302	167,24	167,17	167,21
100	164,06	163,302	167,27	167,21	167,21

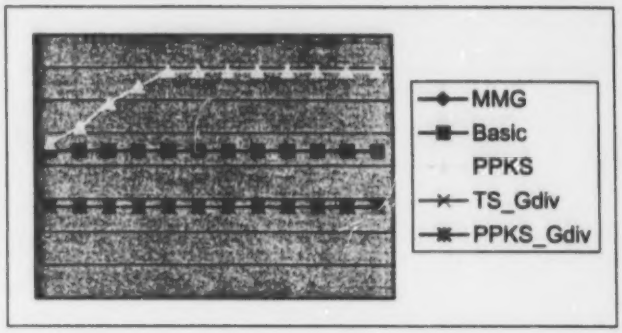


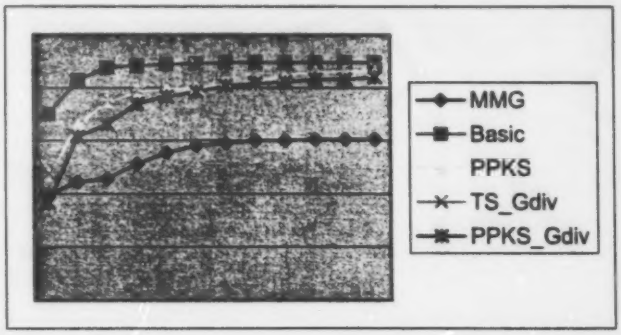




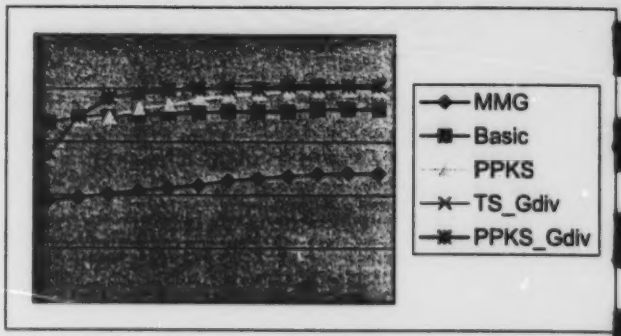
time	MMG Average	Basic Average	PPKS Average	TS_Gdiv Average	PPKS_Gdiv Average
W10	BB =	339.9	*		
1	339,83	339,89	339,89	339,83	339,83
5	339,83	339,89	339,89	339,83	339,83
10	339,83	339,89	339,89	339,83	339,83
20	339,83	339,89	339,89	339,83	339,83
30	339,83	339,89	339,89	339,83	339,83
40	339,83	339,89	339,89	339,83	339,83
50	339,83	339,89	339,89	339,83	339,83
60	339,83	339,89	339,89	339,83	339,83
70	339,83	339,89	339,89	339,83	339,83
80	339,83	339,89	339,89	339,83	339,83
90	339,83	339,89	339,89	339,83	339,83
100	339,83	339,89	339,89	339,83	339,83
W11	BB =	329,46	*		
1	328,55	328,89	328,939	328,55	328,55
5	328,55	328,89	329,037	328,55	328,55
10	328,55	328,89	329,18	328,55	328,55
20	328,55	328,89	329,28	328,55	328,55
30	328,55	328,89	329,38	328,55	328,55
40	328,55	328,89	329,38	328,55	328,55
50	328,55	328,89	329,38	328,55	328,55
60	328,55	328,89	329,38	328,55	328,55
70	328,55	328,89	329,38	328,55	328,55
80	328,55	328,89	329,38	328,55	328,55
90	328,55	328,89	329,38	328,55	328,55
100	328,55	328,89	329,38	328,55	328,55
W12	BB =	213,47			
1	194,92	202,512	196,84	196,84	194,16
5	196,12	205,659	201,48	202,29	200,36
10	196,41	206,928	203,73	203,86	201,55
20	197,82	207,083	204,66	204,75	203,47
30	198,89	207,198	204,85	205,61	204,18
40	199,55	207,293	204,97	206,48	204,59
50	199,74	207,383	205,28	206,54	205,19
60	200,04	207,383	205,40	206,57	205,41
70	200,04	207,387	206,18	206,61	205,49
80	200,13	207,387	206,55	206,62	205,78
90	200,14	207,391	206,56	206,76	205,78
100	200,14	207,391	206,56	206,96	206,01

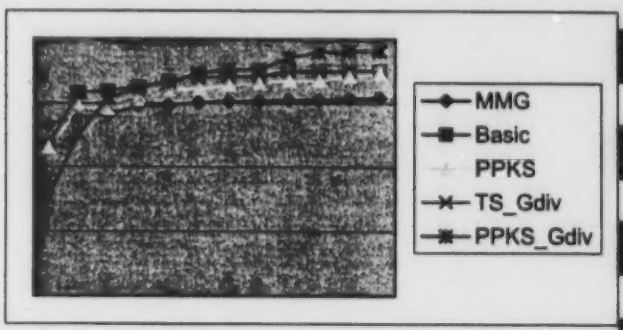


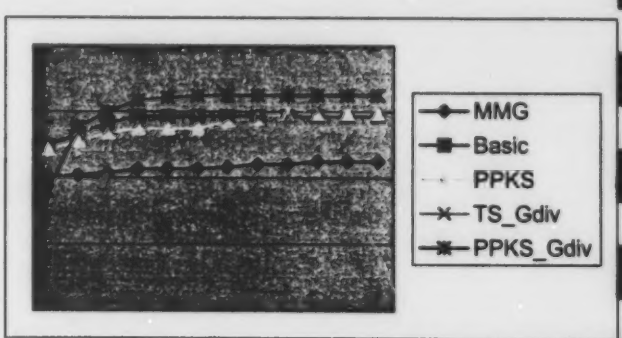




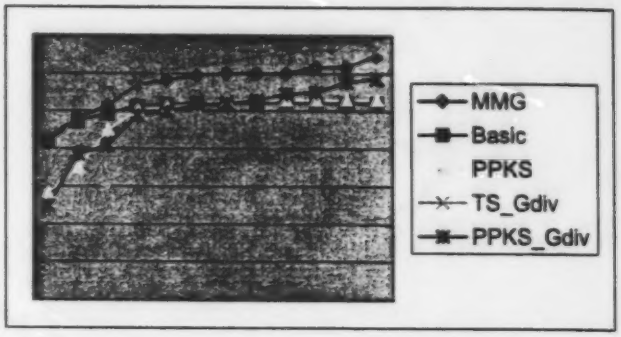
time	MMG Average	Basic Average	PPKS Average	TS_Gdiv Average	PPKS_Gdiv Average
W13	BB =	241,2			
1	219,47	226,784	224,12	224,52	223,97
5	219,77	227,288	227,07	228,22	227,41
10	220,08	227,364	227,34	229,70	229,14
20	220,42	227,551	228,10	229,86	229,58
30	220,70	227,725	228,50	230,04	229,89
40	220,97	227,741	229,10	230,29	230,04
50	221,36	227,867	229,37	230,46	230,05
60	221,55	227,867	229,63	230,47	230,05
70	221,76	227,882	229,83	230,49	230,29
80	222,02	228,072	229,84	230,50	230,29
90	222,06	228,082	229,88	230,50	230,29
100	222,06	228,082	229,88	230,50	230,29
W14	BB =	228,8			
1	218,35	216,724	216,53	214,11	212,57
5	218,85	220,821	219,65	219,04	217,58
10	219,11	220,941	219,54	219,75	220,36
20	219,78	221,287	220,30	220,34	221,16
30	219,79	221,704	221,06	221,19	221,63
40	220,02	222,009	221,11	222,27	222,32
50	220,12	222,057	221,43	222,37	222,64
60	220,14	222,057	221,56	222,81	222,64
70	220,25	222,477	221,78	222,81	223,55
80	220,25	222,477	221,78	222,91	223,98
90	220,35	222,484	222,10	223,00	224,07
100	220,36	222,543	222,34	223,00	224,07
W15	BB =	263,91			
1	250,07	252,35	252,14	246,94	248,91
5	250,17	253,319	252,76	253,51	253,79
10	250,36	254,079	253,39	253,71	255,00
20	250,65	254,639	253,76	254,21	255,88
30	250,71	254,644	253,76	254,47	256,06
40	250,71	254,646	253,77	254,47	256,22
50	250,81	254,658	254,37	254,60	256,24
60	251,02	254,66	254,53	254,60	256,24
70	251,04	254,66	254,83	254,60	256,25
80	251,24	254,66	254,83	255,17	256,25
90	251,24	254,66	254,86	255,48	256,25
100	251,24	254,662	254,86	255,48	256,25

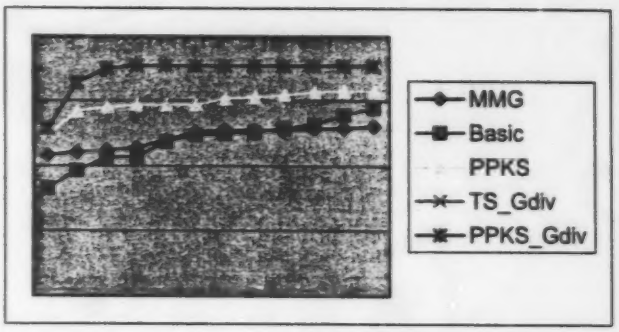


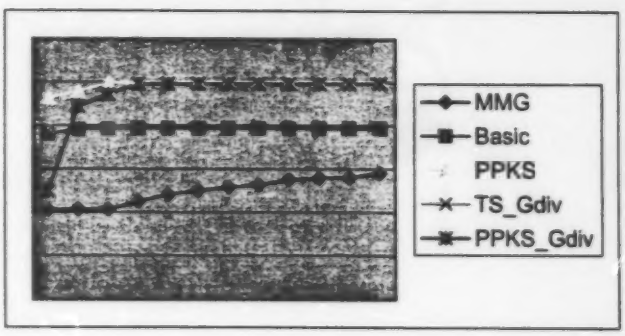




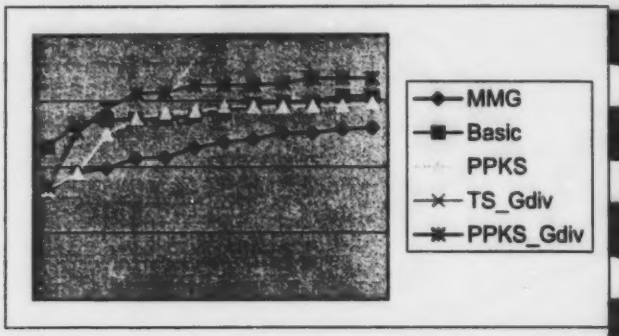
time	MMG Average	Basic Average	PPKS Average	TS_Gdiv Average	PPKS_Gdiv Average
W16	BB =	236,37			
1	202,57	202,382	199,67	200,47	198,94
5	203,64	203,447	201,00	204,53	201,71
10	204,25	203,862	203,03	205,23	202,14
20	205,40	204,259	204,08	205,59	203,79
30	205,63	204,371	204,26	206,14	203,96
40	205,89	204,406	204,32	206,37	204,35
50	206,00	204,406	204,56	206,57	204,39
60	206,00	204,418	204,60	206,91	204,58
70	206,00	204,418	204,62	207,04	204,91
80	206,35	204,425	204,62	207,26	205,01
90	206,35	204,425	204,63	207,26	205,53
100	206,95	204,439	204,63	207,26	205,73
W17	BB =	198,53			
1	175,83	173,281	177,48	177,17	177,87
5	176,02	174,627	179,23	181,63	181,41
10	176,19	175,65	179,53	182,25	182,52
20	176,38	175,674	179,69	182,70	182,71
30	176,98	176,846	179,71	182,71	182,72
40	177,62	177,323	179,76	182,71	182,72
50	177,74	177,484	180,19	182,71	182,72
60	177,74	177,542	180,32	182,71	182,72
70	177,74	177,845	180,49	182,72	182,72
80	177,87	178,184	180,70	182,72	182,72
90	177,87	178,909	180,82	182,72	182,73
100	178,05	179,423	180,82	182,72	182,73
W18	BB =	247,39			
1	205,26	213,944	218,26	214,72	207,38
5	205,32	214,592	219,03	217,52	217,21
10	205,32	214,656	220,01	218,14	218,56
20	206,21	214,696	220,04	219,04	219,82
30	207,00	214,701	220,04	219,39	219,83
40	207,50	214,708	220,05	219,83	219,84
50	207,86	214,708	220,05	219,83	219,84
60	208,10	214,708	220,06	219,84	219,84
70	208,69	214,709	220,06	219,84	219,84
80	208,90	214,709	220,06	219,84	219,85
90	208,99	214,71	220,06	219,84	219,85
100	209,51	214,71	220,07	219,85	219,86

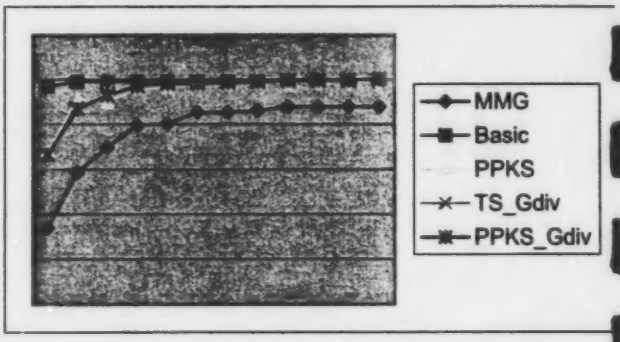


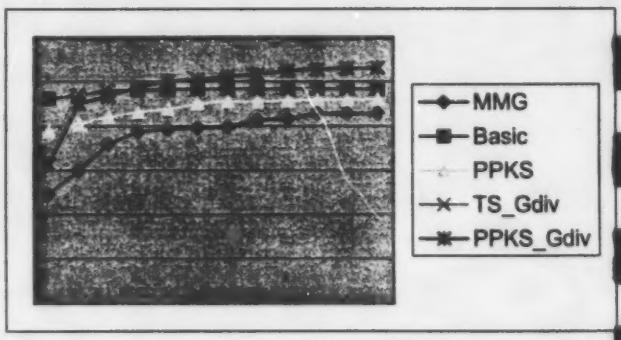




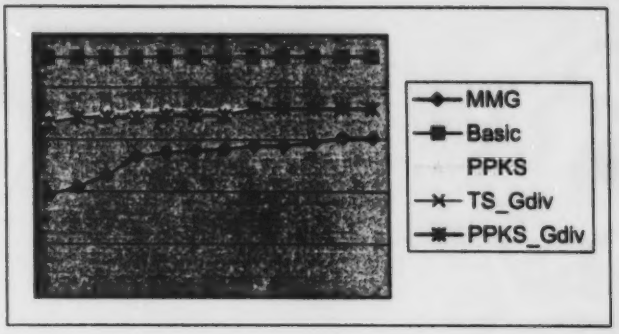
time	MMG Average	Basic Average	PPKS Average	TS_Gdiv Average	PPKS_Gdiv Average
W19	BB =	205,18			
1	178,09	181,54	178,25	178,80	178,52
5	179,61	183,356	179,56	183,69	182,60
10	179,88	183,514	182,51	183,94	184,45
20	180,62	183,612	183,65	184,76	185,55
30	180,72	183,619	184,11	186,23	185,65
40	181,44	183,855	184,21	186,98	186,13
50	181,87	184,085	184,55	187,04	186,20
60	182,08	184,725	184,55	187,07	186,20
70	182,53	184,856	184,56	187,26	186,32
80	182,53	184,914	184,59	187,26	186,70
90	182,84	185,313	184,66	187,35	186,70
100	182,92	185,32	184,84	187,46	186,70
W20	BB =	256,59			
1	237,39	243,7	240,81	240,02	240,65
5	239,87	243,891	243,16	242,76	242,69
10	240,97	243,895	243,01	243,54	243,31
20	241,95	243,899	243,66	243,75	243,69
30	242,03	243,9	243,66	243,81	243,82
40	242,55	243,9	243,83	243,81	243,84
50	242,55	243,9	243,83	243,82	243,88
60	242,61	243,9	243,85	243,82	243,90
70	242,78	243,9	243,85	243,84	243,98
80	242,79	243,9	243,88	243,85	243,98
90	242,79	243,9	243,89	243,85	243,98
100	242,79	243,9	243,89	243,85	244,03
W21	BB =	179,32			
1	156,91	161,231	159,77	158,85	158,26
5	157,96	161,385	160,09	161,58	161,00
10	159,17	161,585	160,42	161,92	161,32
20	159,73	161,604	160,63	162,25	161,74
30	159,85	161,645	160,80	162,36	162,03
40	159,92	161,651	161,01	162,38	162,07
50	159,96	161,654	161,12	162,38	162,23
60	160,26	161,654	161,12	162,48	162,31
70	160,33	161,654	161,14	162,62	162,48
80	160,57	161,654	161,19	162,66	162,54
90	160,57	161,654	161,19	162,70	162,60
100	160,57	161,654	161,23	162,93	162,62

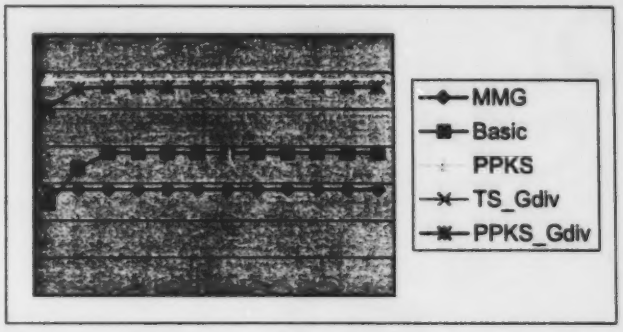




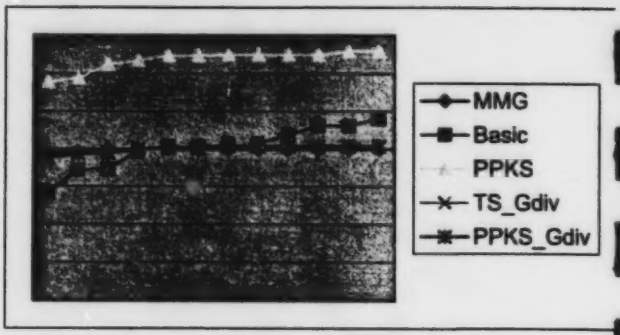


time	MMG Average	Basic Average	PPKS Average	TS_Gdiv Average	PPKS_Gdiv Average
W22	BB =	278,14	*		
1	264,70	277,804	271,97	273,09	271,55
5	265,45	277,818	272,12	273,93	272,06
10	266,60	277,844	272,72	273,93	272,13
20	268,34	277,848	272,72	274,45	272,13
30	268,74	277,887	272,87	274,46	272,27
40	268,88	277,889	272,88	274,46	272,27
50	269,05	277,902	272,89	274,98	272,30
60	269,41	277,911	272,89	274,99	272,93
70	269,41	277,92	272,89	274,99	272,94
80	269,69	277,93	272,89	274,99	272,94
90	270,14	277,93	272,91	274,99	272,94
100	270,14	277,93	272,92	274,99	272,94
W23	BB =	196,92	*		
1	193,83	193,458	196,80	195,89	196,05
5	193,83	194,421	196,80	196,56	196,55
10	193,83	194,826	196,81	196,61	196,60
20	193,83	194,826	196,81	196,61	196,60
30	193,83	194,826	196,81	196,61	196,60
40	193,83	194,826	196,81	196,61	196,60
50	193,83	194,826	196,81	196,61	196,60
60	193,83	194,826	196,81	196,61	196,60
70	193,83	194,826	196,81	196,61	196,60
80	193,83	194,826	196,81	196,61	196,60
90	193,83	194,826	196,81	196,61	196,60
100	193,83	194,826	196,81	196,61	196,60

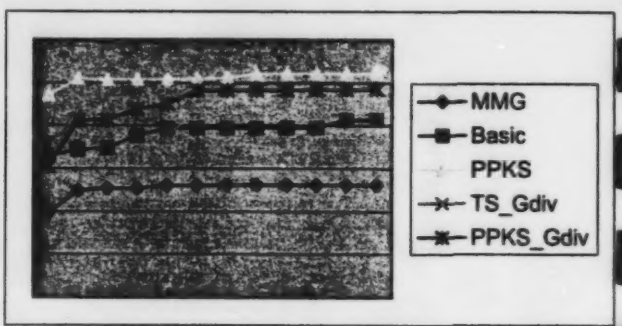




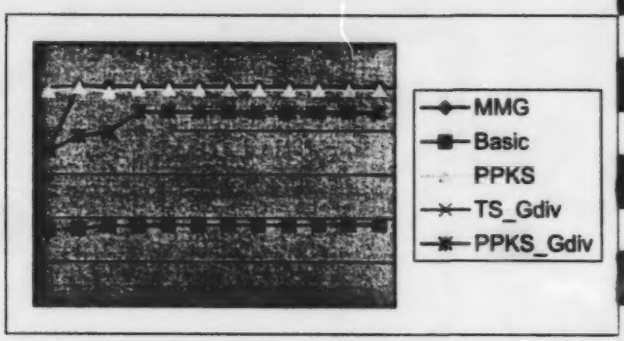
time	MMG Average	Basic Average	PPKS Average	TS_Gdiv Average	PPKS_Gdiv Average
S1	BB =	130,27			
1	127,94	127,415	128,89	127,78	127,87
5	127,96	127,718	128,94	127,95	127,93
10	127,98	127,73	129,12	127,97	127,95
20	127,99	127,966	129,17	127,97	128,00
30	128,00	128,037	129,24	128,00	128,01
40	128,00	128,037	129,24	128,05	128,01
50	128,00	128,07	129,24	128,07	128,03
60	128,00	128,07	129,24	128,07	128,03
70	128,00	128,181	129,25	128,09	128,03
80	128,00	128,301	129,25	128,09	128,03
90	128,00	128,301	129,29	128,09	128,03
100	128,00	128,399	129,29	128,09	128,03



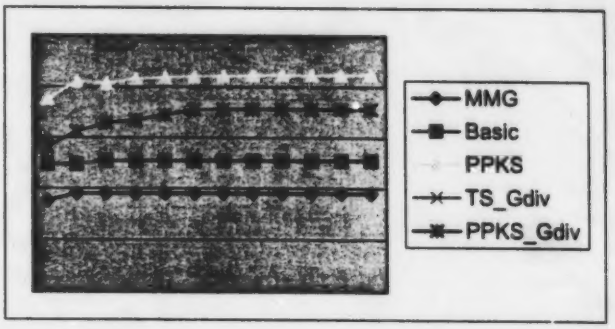
time	MMG Average	Basic Average	PPKS Average	TS_Gdiv Average	PPKS_Gdiv Average
S2	BB =	133,63			
1	124,96	126,296	127,71	126,56	126,04
5	125,48	126,436	128,10	127,20	127,12
10	125,53	126,47	128,08	127,68	127,14
20	125,54	126,76	128,09	127,80	127,38
30	125,61	126,938	128,09	127,80	127,51
40	125,61	126,938	128,11	127,82	127,82
50	125,62	126,938	128,11	127,83	127,83
60	125,62	126,938	128,20	127,86	127,83
70	125,62	126,954	128,22	127,86	127,83
80	125,62	126,954	128,22	127,87	127,83
90	125,62	127,132	128,22	127,87	127,83
100	125,62	127,132	128,22	127,89	127,84

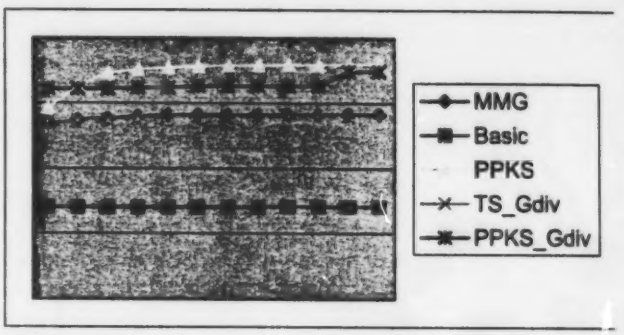


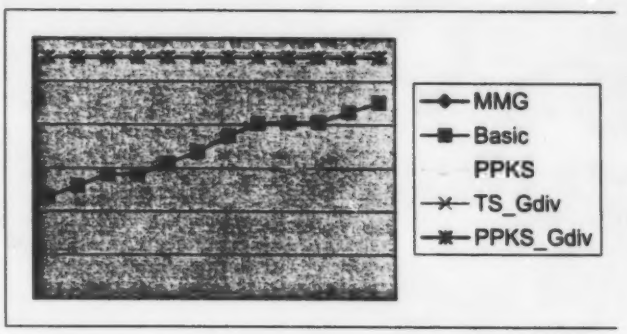
time	MMG Average	Basic Average	PPKS Average	TS_Gdiv Average	PPKS_Gdiv Average
S3	BB =	139,43	*		
1	137,54	135,73	138,90	137,93	137,53
5	139,01	135,757	138,99	138,02	137,87
10	139,01	135,813	138,88	138,02	137,96
20	139,01	135,822	138,95	138,02	138,43
30	139,01	135,85	138,95	138,02	138,43
40	139,01	135,85	138,95	138,33	138,43
50	139,01	135,85	138,95	138,33	138,43
60	139,01	135,85	138,95	138,33	138,43
70	139,01	135,85	138,95	138,33	138,43
80	139,01	135,85	138,95	138,33	138,43
90	139,01	135,851	138,95	138,33	138,43
100	139,01	135,851	138,95	138,33	138,43



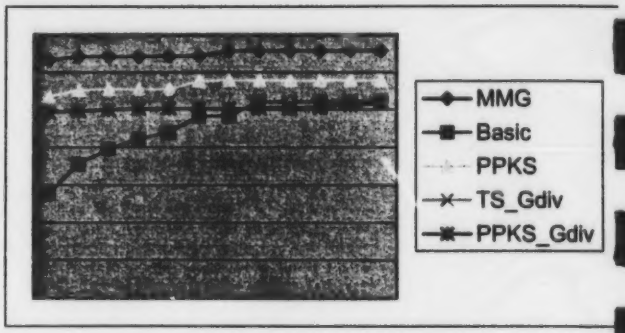
time	MMG Average	Basic Average	PPKS Average	TS_Gdiv Average	PPKS_Gdiv Average
S4	BB =	122,39	*		
1	119,76	120,477	121,75	121,07	120,81
5	119,90	120,477	122,10	121,54	121,12
10	119,90	120,54	122,06	121,66	121,28
20	119,90	120,54	122,15	121,68	121,34
30	119,90	120,54	122,19	121,75	121,45
40	119,90	120,54	122,20	121,75	121,55
50	119,90	120,54	122,21	121,75	121,55
60	119,90	120,54	122,23	121,75	121,55
70	119,90	120,54	122,23	121,75	121,55
80	119,90	120,54	122,23	121,75	121,56
90	119,90	120,54	122,23	121,75	121,56
100	119,90	120,54	122,24	121,75	121,56
S5	BB =	121,91	*		
1	121,35	120,7	121,46	121,36	121,61
5	121,37	120,7	121,64	121,36	121,61
10	121,38	120,701	121,73	121,36	121,61
20	121,40	120,701	121,75	121,38	121,62
30	121,41	120,701	121,77	121,39	121,62
40	121,41	120,702	121,78	121,39	121,63
50	121,41	120,702	121,79	121,39	121,63
60	121,41	120,703	121,79	121,40	121,63
70	121,41	120,704	121,80	121,40	121,63
80	121,41	120,704	121,80	121,40	121,63
90	121,41	120,704	121,80	121,40	121,74
100	121,41	120,704	121,80	121,40	121,74
S6	BB =	107,82			
1	105,12	98,665	105,42	105,13	105,13
5	105,13	99,184	105,44	105,13	105,13
10	105,13	99,703	105,45	105,13	105,13
20	105,13	99,703	105,45	105,13	105,13
30	105,13	100,223	105,45	105,14	105,13
40	105,13	100,742	105,45	105,14	105,13
50	105,13	101,484	105,45	105,14	105,13
60	105,13	102,06	105,45	105,14	105,13
70	105,13	102,12	105,45	105,14	105,13
80	105,13	102,12	105,45	105,14	105,13
90	105,13	102,579	105,45	105,14	105,13
100	105,13	103,038	105,45	105,14	105,13

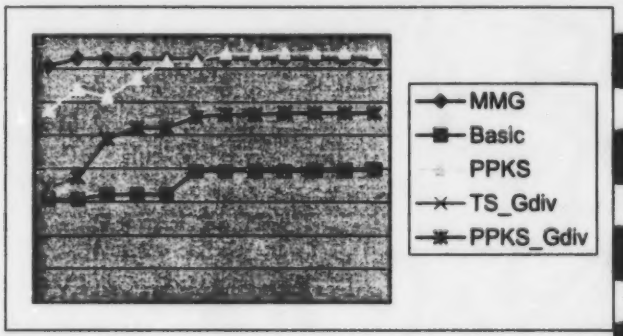


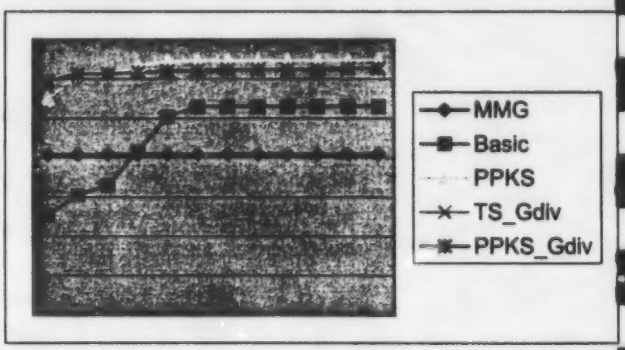




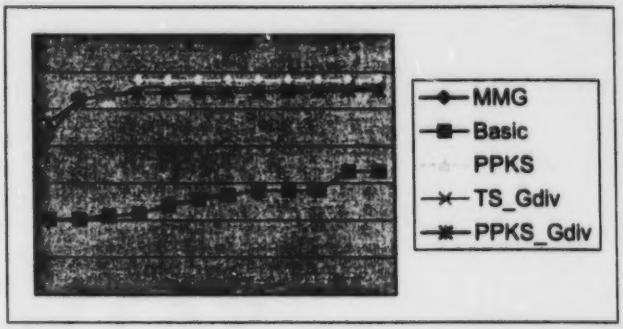
time	MMG Average	Basic Average	PPKS Average	TS_Gdiv Average	PPKS_Gdiv Average
S7	BB =	139,62			
1	136,32	132,759	135,36	134,88	134,93
5	136,42	133,557	135,53	135,17	134,98
10	136,42	133,95	135,56	135,20	135,01
20	136,42	134,231	135,56	135,23	135,03
30	136,43	134,399	135,57	135,24	135,03
40	136,44	134,834	135,78	135,25	135,04
50	136,56	134,842	135,78	135,26	135,04
60	136,56	135,217	135,78	135,28	135,05
70	136,57	135,217	135,78	135,28	135,05
80	136,57	135,222	135,78	135,29	135,07
90	136,57	135,222	135,78	135,29	135,12
100	136,57	135,228	135,78	135,29	135,18
S8	BB =	143,41	*		
1	134,04	132,034	133,38	132,51	132,21
5	134,15	132,034	133,69	132,96	132,36
10	134,15	132,108	133,57	132,99	132,94
20	134,15	132,108	133,86	133,25	133,11
30	134,15	132,108	134,11	133,31	133,11
40	134,15	132,459	134,12	133,42	133,27
50	134,15	132,459	134,23	133,56	133,31
60	134,15	132,459	134,23	133,56	133,31
70	134,15	132,459	134,23	133,69	133,33
80	134,15	132,463	134,24	133,69	133,33
90	134,15	132,467	134,24	133,69	133,33
100	134,15	132,483	134,25	133,70	133,33
S9	BB =	144,65	*		
1	142,08	140,527	143,51	143,79	143,95
5	142,09	141,033	144,11	143,85	144,13
10	142,09	141,291	144,08	143,86	144,13
20	142,09	142,17	144,35	143,86	144,17
30	142,09	143,054	144,47	143,88	144,20
40	142,09	143,307	144,49	144,09	144,23
50	142,09	143,307	144,49	144,20	144,25
60	142,09	143,307	144,49	144,20	144,25
70	142,09	143,307	144,49	144,20	144,25
80	142,09	143,307	144,49	144,20	144,25
90	142,09	143,307	144,49	144,20	144,27
100	142,09	143,307	144,49	144,20	144,27

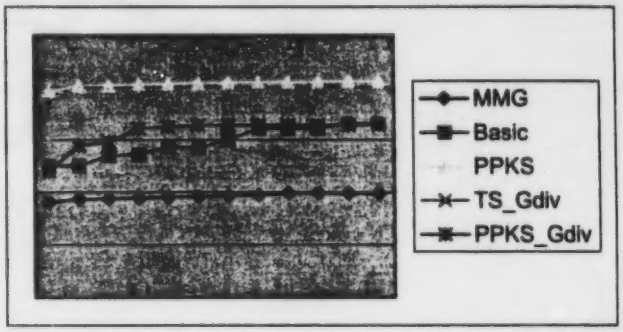


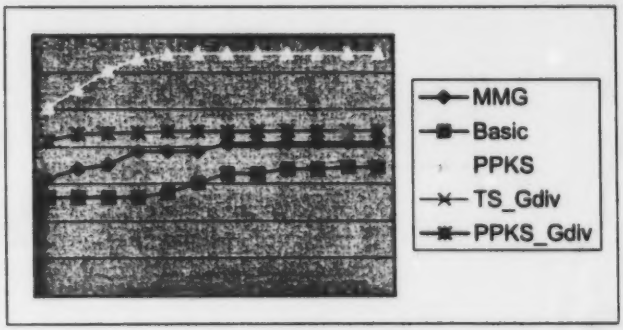




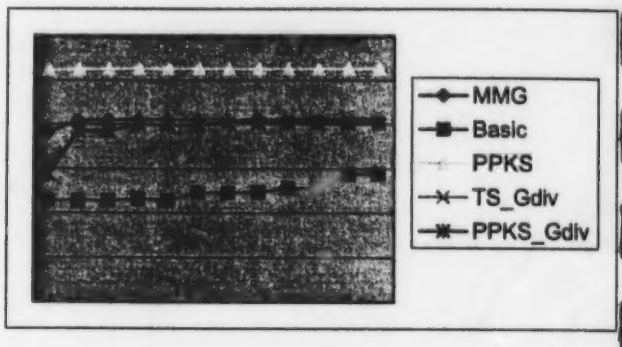
time	MMG Average	Basic Average	PPKS Average	TS_Gdiv Average	PPKS_Gdiv Average
S10	BB =	139,1			
1	135,26	133,029	135,73	135,27	135,64
5	136,27	133,029	136,26	136,02	136,24
10	136,32	133,113	136,26	136,41	136,42
20	136,59	133,179	136,78	136,58	136,44
30	136,59	133,389	136,78	136,58	136,45
40	136,60	133,564	136,78	136,59	136,46
50	136,60	133,676	136,78	136,61	136,46
60	136,62	133,801	136,78	136,61	136,48
70	136,62	133,801	136,79	136,61	136,50
80	136,62	133,801	136,79	136,61	136,50
90	136,62	134,324	136,79	136,62	136,50
100	136,62	134,324	136,79	136,62	136,50
S11	BB =	134,42	*		
1	131,81	132,496	133,94	132,99	132,40
5	131,86	132,496	134,04	133,36	132,89
10	131,87	132,694	134,04	133,48	132,96
20	131,89	132,713	134,07	133,53	133,21
30	131,91	132,867	134,08	133,53	133,25
40	131,91	132,841	134,09	133,53	133,28
50	131,92	133,003	134,10	133,54	133,28
60	131,95	133,218	134,11	133,54	133,28
70	131,98	133,218	134,11	133,54	133,28
80	131,99	133,218	134,12	133,54	133,28
90	131,99	133,314	134,12	133,54	133,28
100	131,99	133,314	134,12	133,54	133,29
S12	BB =	146,65	*		
1	143,18	142,64	145,07	143,37	144,17
5	143,39	142,64	145,57	144,21	144,35
10	143,53	142,64	146,07	144,48	144,37
20	143,87	142,64	146,37	144,81	144,38
30	143,87	142,757	146,49	144,81	144,41
40	143,88	143,015	146,51	144,81	144,41
50	144,09	143,282	146,51	144,87	144,41
60	144,09	143,282	146,51	144,87	144,41
70	144,10	143,407	146,51	144,87	144,42
80	144,10	143,407	146,51	144,87	144,42
90	144,10	143,442	146,51	144,87	144,42
100	144,10	143,442	146,51	144,87	144,42

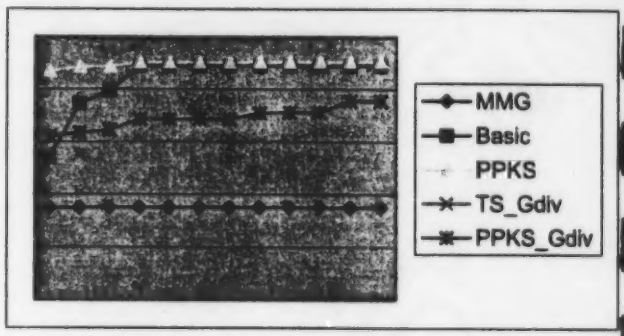


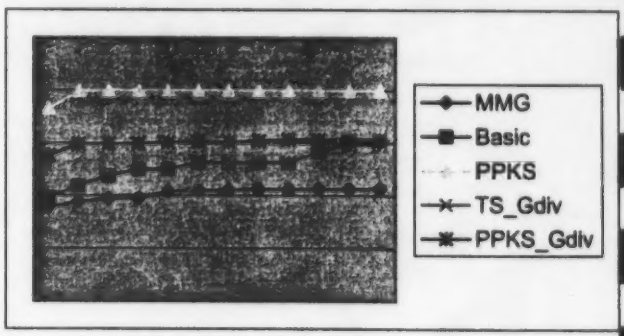




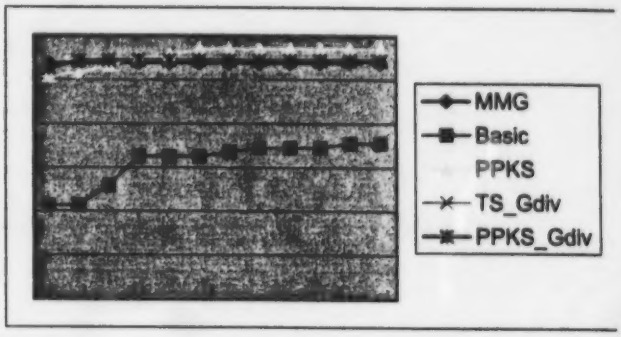
time	MMG Average	Basic Average	PPKS Average	TS_Gdiv Average	PPKS_Gdiv Average
S13	BB =	153,11			
1	131,63	128,565	134,48	130,38	129,70
5	132,16	128,566	134,48	131,33	131,69
10	132,20	128,566	134,51	131,59	131,71
20	132,20	128,629	134,51	132,05	131,97
30	132,20	128,566	134,52	132,19	131,97
40	132,20	128,908	134,52	132,20	131,97
50	132,20	128,908	134,52	132,21	132,04
60	132,22	128,908	134,53	132,37	132,04
70	132,24	129,128	134,53	132,40	132,04
80	132,24	129,128	134,53	132,40	132,04
90	132,24	129,752	134,53	132,40	132,04
100	132,24	129,752	134,53	132,40	132,04
S14	BB =	135,53	*		
1	132,74	133,684	135,35	133,61	134,10
5	132,75	134,725	135,44	133,75	134,17
10	132,75	134,968	135,45	133,99	134,22
20	132,75	135,445	135,53	134,03	134,43
30	132,75	135,445	135,53	134,03	134,43
40	132,75	135,445	135,53	134,03	134,44
50	132,75	135,445	135,53	134,03	134,44
60	132,75	135,445	135,53	134,03	134,55
70	132,75	135,445	135,53	134,03	134,56
80	132,75	135,445	135,53	134,24	134,56
90	132,75	135,445	135,53	134,24	134,77
100	132,75	135,445	135,53	134,24	134,77
S15	BB =	117,3			
1	113,76	113,957	115,65	114,50	114,76
5	113,89	114,184	115,99	114,69	114,97
10	113,94	114,369	116,00	114,70	114,99
20	113,95	114,498	116,00	114,75	114,99
30	114,08	114,498	116,01	114,75	115,01
40	114,10	114,631	116,01	114,77	115,04
50	114,11	114,631	116,01	114,77	115,04
60	114,13	114,631	116,01	114,80	115,04
70	114,13	114,631	116,01	114,81	115,07
80	114,13	114,848	116,01	114,81	115,07
90	114,13	114,977	116,01	114,81	115,07
100	114,14	114,977	116,01	114,83	115,07

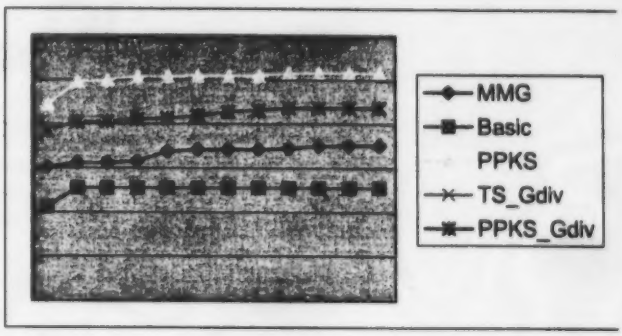


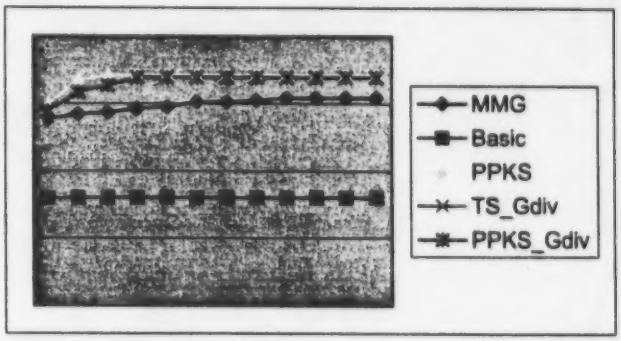




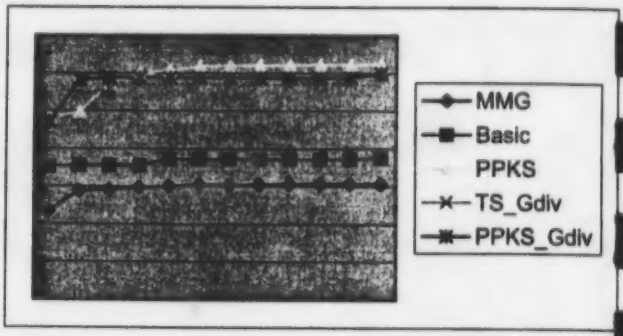
time	MMG Average	Basic Average	PPKS Average	TS_Gdiv Average	PPKS_Gdiv Average
S16	BB =	110,05			
1	105,37	102,138	105,08	104,45	105,24
5	105,37	102,138	105,16	104,94	105,42
10	105,37	102,587	105,32	104,95	105,43
20	105,37	103,245	105,48	104,96	105,43
30	105,37	103,245	105,55	104,96	105,43
40	105,37	103,245	105,69	104,96	105,43
50	105,37	103,347	105,70	104,98	105,43
60	105,37	103,449	105,70	105,01	105,43
70	105,37	103,449	105,70	105,17	105,43
80	105,37	103,449	105,70	105,17	105,43
90	105,37	103,551	105,70	105,30	105,43
100	105,37	103,551	105,70	105,42	105,43
S17	BB =	150,36			
1	147,02	146,105	148,45	147,78	147,85
5	147,12	146,548	148,97	148,17	148,04
10	147,12	146,548	148,99	148,19	148,06
20	147,16	146,548	149,08	148,32	148,10
30	147,38	146,549	149,08	148,45	148,13
40	147,43	146,549	149,10	148,51	148,19
50	147,43	146,549	149,10	148,51	148,28
60	147,43	146,549	149,10	148,51	148,31
70	147,45	146,549	149,16	148,51	148,37
80	147,50	146,549	149,16	148,51	148,37
90	147,52	146,549	149,16	148,51	148,37
100	147,52	146,549	149,16	148,53	148,37
S18	BB =	132,21	*		
1	128,84	122,932	130,02	129,15	129,44
5	129,19	122,994	131,81	130,79	130,84
10	129,31	122,994	132,14	131,37	131,36
20	129,61	122,994	132,15	131,67	131,95
30	129,77	122,994	132,15	131,85	131,95
40	130,12	122,994	132,17	131,85	131,96
50	130,12	122,994	132,17	131,88	131,96
60	130,26	122,994	132,17	131,89	131,96
70	130,42	122,994	132,17	131,89	131,96
80	130,42	122,994	132,17	131,89	131,96
90	130,42	122,994	132,17	131,89	131,96
100	130,42	122,994	132,17	131,89	131,96

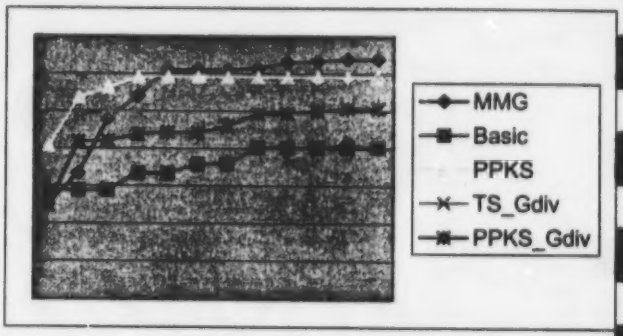






time	MMG Average	Basic Average	PPKS Average	TS_Gdiv Average	PPKS_Gdiv Average
S19	BB =	130,26	*		
1	126,36	127,529	128,82	128,64	128,72
5	126,88	127,549	129,00	129,47	129,82
10	126,91	127,551	129,75	129,58	129,87
20	126,96	127,551	129,97	129,71	129,87
30	127,00	127,722	130,07	129,71	129,87
40	127,04	127,722	130,19	129,71	129,88
50	127,04	127,722	130,20	129,71	129,88
60	127,04	127,722	130,20	129,71	129,88
70	127,05	127,722	130,23	129,71	129,88
80	127,05	127,722	130,24	129,71	129,88
90	127,05	127,722	130,24	129,71	129,88
100	127,05	127,76	130,24	129,71	129,99
S20	BB =	143,74			
1	129,75	129,942	130,54	130,08	129,73
5	130,16	129,942	131,18	130,94	130,59
10	130,90	129,942	131,32	131,00	130,60
20	131,21	130,169	131,46	131,01	130,69
30	131,55	130,169	131,46	131,20	130,71
40	131,55	130,288	131,46	131,20	130,71
50	131,55	130,288	131,46	131,21	130,80
60	131,58	130,515	131,46	131,27	130,97
70	131,65	130,515	131,46	131,27	130,97
80	131,67	130,515	131,46	131,31	131,00
90	131,70	130,515	131,46	131,34	131,04
100	131,70	130,515	131,46	131,46	131,04





Appendix 2

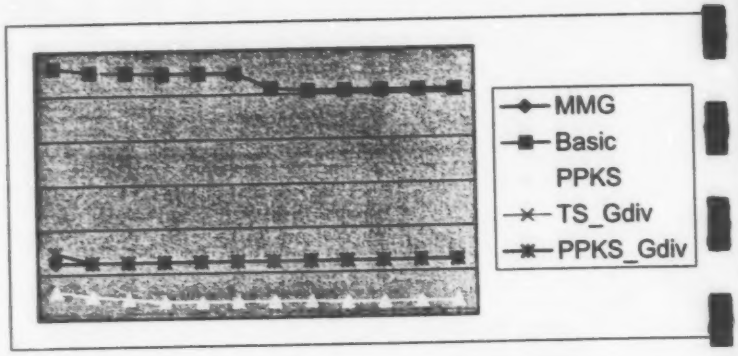
This appendix includes the detailed gap results for the five selected meta-heuristics: the memory-enhanced, randomized greedy constructive procedure MMG, the **Basic** tabu search, the tabu search approach that makes use of long-term request-satisfaction memories and probabilistic node potential revenue **PPKS**, as well as the two hybrids **TS_Gdiv** and **PPKS_Gdiv** that combine the respective tabu search approaches with the MMG method as a diversification device.

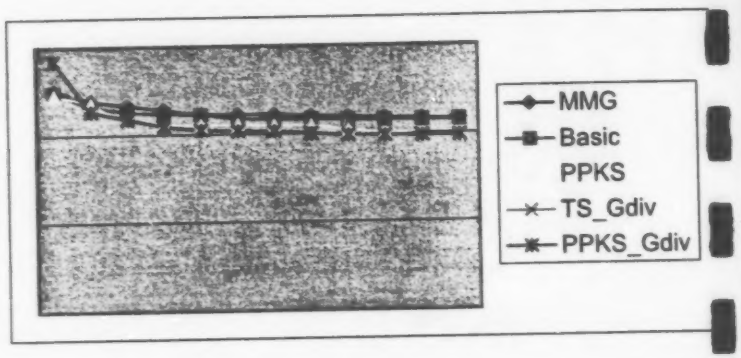
Results are given for twelve levels of computational efforts: 1, 5, 10, 20, 30, 40, 50, 60, 70, 80, 90, and 100 seconds of CPU time on a Pentium II 233 Mhz PC.

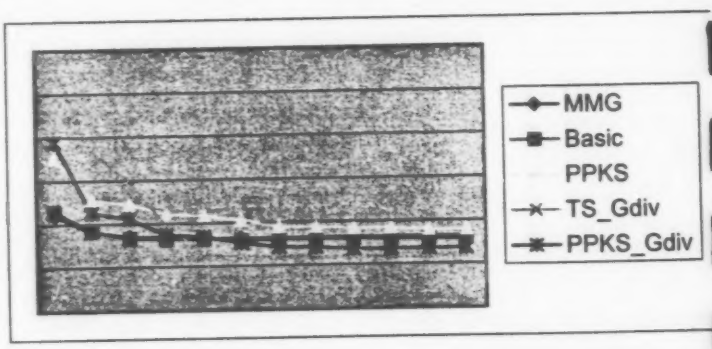
We indicate the gap in percentage of each solution value relative to the best upper bound identified by CPLEX after 100 hours of CPU time for each problem instance and each level of computational effort. The evolution of these gaps with the CPU time is also graphically displayed.

The best upper bound identified by the branch-and-bound method of CPLEX 6.6 is also shown. A * identifies an optimal solution. The branch-and-bound algorithm has been run for 10 hours (wall-clock time) of computing time on 10 of the 64 processors of a SUN E10000 computer (with 64 Gb of RAM memory and 400 Mhz processors running under Solaris 2.7).

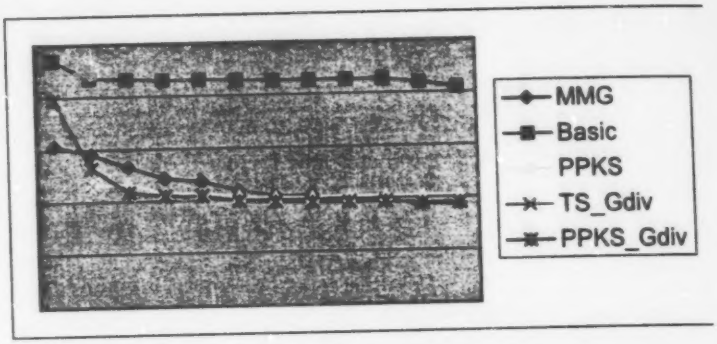
time	MMG Mean	Basic Mean	PPKS Mean	TS_Gdiv Mean	PPKS_Gdiv Mean
W1	BB =	278,96	*		
	0,13	0,56	0,06	0,27	0,15
	0,13	0,55	0,05	0,13	0,13
	0,13	0,55	0,04	0,13	0,13
	0,13	0,55	0,04	0,13	0,13
	0,13	0,55	0,04	0,13	0,13
	0,13	0,55	0,04	0,13	0,13
	0,13	0,51	0,04	0,13	0,13
	0,13	0,51	0,04	0,13	0,13
	0,13	0,51	0,03	0,13	0,13
	0,13	0,51	0,03	0,13	0,13
	0,13	0,51	0,03	0,13	0,13
	0,13	0,51	0,03	0,13	0,13
W2	BB =	217,7			
	12,54	12,41	12,27	13,16	14,20
	11,90	11,58	11,77	11,39	11,33
	11,65	11,29	10,66	10,72	10,88
	11,39	11,16	10,58	10,33	10,46
	11,16	11,11	10,53	10,27	10,25
	11,13	10,85	10,52	10,18	10,19
	11,12	10,82	10,51	10,16	10,16
	11,02	10,82	10,46	10,15	9,99
	10,94	10,76	10,20	9,97	9,89
	10,90	10,75	10,02	9,96	9,87
	10,82	10,75	9,89	9,89	9,87
	10,76	10,75	9,89	9,89	9,87
W3	BB =	211,15			
	2,20	2,29	3,54	5,68	3,89
	1,81	1,84	2,56	3,67	2,26
	1,69	1,65	2,46	2,97	2,11
	1,69	1,64	2,22	1,97	1,73
	1,63	1,62	2,17	1,27	1,68
	1,63	1,60	2,07	1,21	1,53
	1,63	1,59	1,89	1,21	1,43
	1,63	1,59	1,88	1,21	1,41
	1,62	1,59	1,84	1,20	1,36
	1,62	1,58	1,84	1,16	1,36
	1,62	1,58	1,79	1,16	1,35
	1,62	1,57	1,78	1,16	1,35

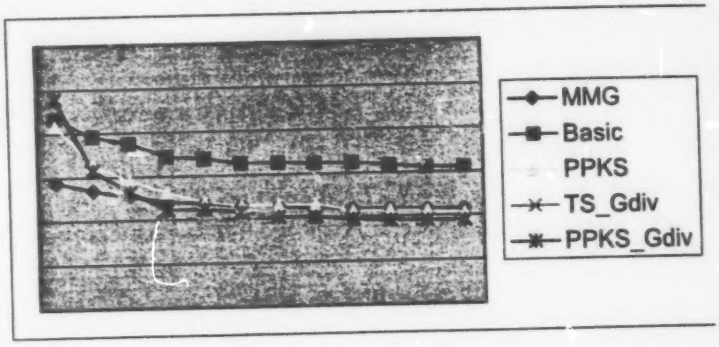


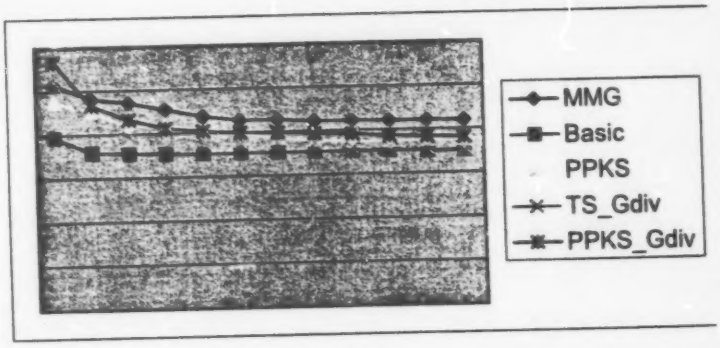




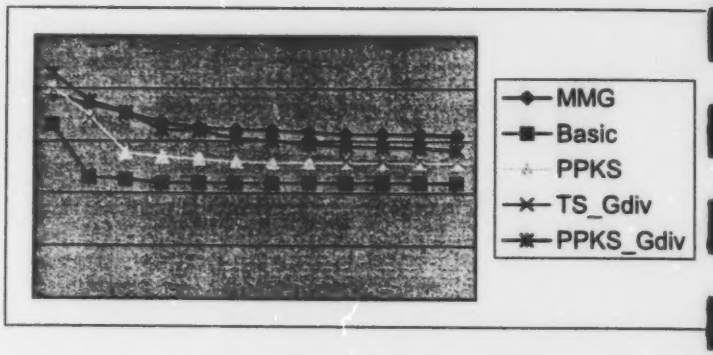
time	MMG Mean	Basic Mean	PPKS Mean	TS_Gdiv Mean	PPKS_Gdiv Mean
W4	BB =	227,7			
	6,15	9,40	7,65	7,69	7,90
	5,82	8,70	4,64	4,61	5,28
	5,30	8,65	4,69	4,21	4,35
	4,87	8,60	4,29	4,06	4,19
	4,73	8,60	4,27	4,00	4,17
	4,38	8,59	4,18	3,98	3,96
	4,12	8,53	4,11	3,88	3,92
	4,12	8,53	4,09	3,85	3,88
	3,96	8,53	4,08	3,85	3,88
	3,87	8,53	4,08	3,85	3,88
	3,74	8,38	4,08	3,76	3,78
	3,74	8,19	4,07	3,76	3,78
W5	BB =	227,26			
	5,76	8,68	8,34	9,13	9,57
	5,36	7,84	6,56	6,36	6,28
	5,08	7,51	5,80	4,70	5,29
	4,82	6,90	5,27	4,13	4,47
	4,72	6,76	4,77	4,02	4,46
	4,60	6,53	4,59	3,86	4,31
	4,54	6,53	4,58	3,74	4,06
	4,43	6,48	4,58	3,61	4,02
	4,36	6,48	4,29	3,54	3,91
	4,36	6,36	4,27	3,54	3,81
	4,32	6,23	4,24	3,46	3,80
	4,31	6,22	4,12	3,46	3,76
W6	BB =	245,96			
	10,27	7,88	9,50	10,44	11,32
	9,65	7,16	9,04	8,46	9,34
	9,42	7,09	8,37	7,89	8,66
	9,08	7,06	8,32	7,87	8,22
	8,76	7,05	8,12	7,57	8,08
	8,55	7,04	7,94	7,54	7,97
	8,55	7,04	7,68	7,48	7,96
	8,44	7,03	7,55	7,44	7,86
	8,44	7,03	7,49	7,18	7,81
	8,43	7,03	7,48	7,18	7,70
	8,42	7,02	7,48	7,11	7,67
	8,40	7,02	7,48	7,10	7,61

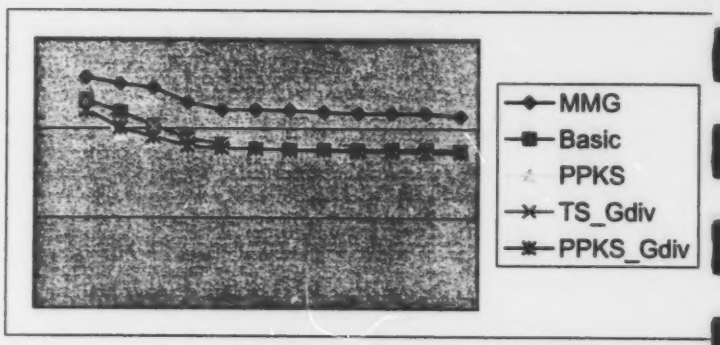


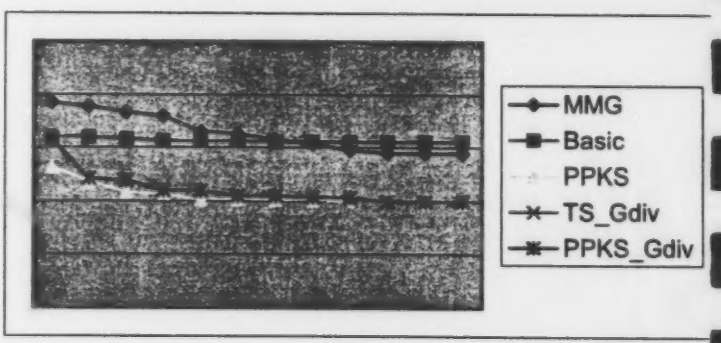




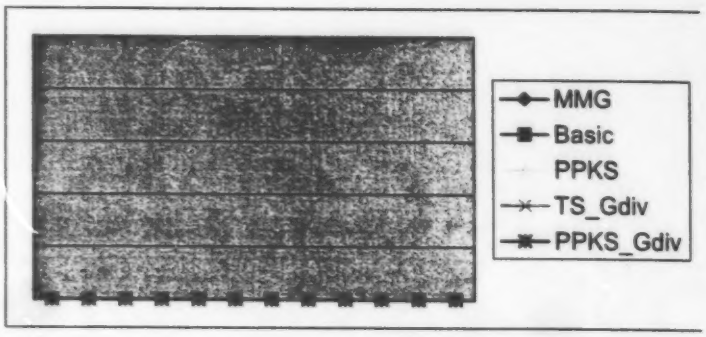
time	MMG Mean	Basic Mean	PPKS Mean	TS_Gdiv Mean	PPKS_Gdiv Mean
W7	BB =	228,18			
	7,74	6,63	8,02	7,93	8,57
	7,40	4,64	6,96	6,86	7,53
	7,07	4,54	5,56	6,54	7,08
	6,70	4,41	5,41	6,18	6,42
	6,51	4,40	5,32	5,90	6,39
	6,44	4,40	5,21	5,76	6,10
	6,40	4,40	5,20	5,50	6,00
	6,33	4,39	5,19	5,47	5,87
	6,26	4,39	5,19	5,27	5,80
	6,26	4,39	5,19	5,26	5,75
	6,25	4,39	5,19	5,26	5,74
	6,20	4,39	5,17	5,25	5,65
W8	BB =	272,74			
	12,84	11,57	11,03	11,57	10,99
	12,49	10,91	9,69	10,36	10,02
	12,26	10,14	9,45	10,04	9,53
	11,46	9,44	8,97	9,30	9,02
	11,03	9,10	8,86	9,04	8,87
	11,03	8,84	8,78	8,93	8,84
	11,01	8,83	8,78	8,83	8,79
	10,92	8,83	8,76	8,77	8,79
	10,83	8,81	8,75	8,68	8,73
	10,83	8,81	8,75	8,67	8,73
	10,83	8,80	8,75	8,57	8,65
	10,70	8,76	8,67	8,57	8,65
W9	BB =	174,12			
	7,69	6,41	5,28	6,85	6,29
	7,54	6,39	4,73	5,31	4,84
	7,33	6,33	4,34	5,18	4,76
	7,17	6,25	4,25	4,57	4,41
	6,61	6,24	4,05	4,54	4,38
	6,53	6,23	4,05	4,29	4,18
	6,31	6,22	4,04	4,29	4,18
	6,14	6,21	4,03	4,29	4,15
	5,91	6,21	4,02	4,02	4,15
	5,78	6,21	4,01	4,02	3,97
	5,78	6,21	3,95	3,99	3,97
	5,78	6,21	3,94	3,97	3,97

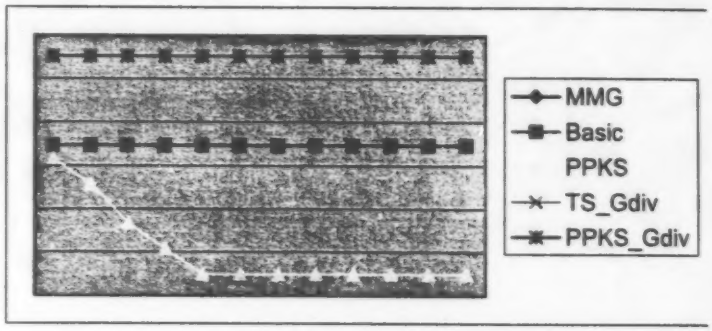


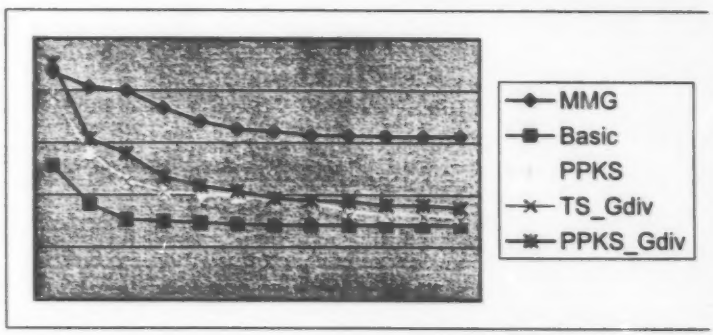




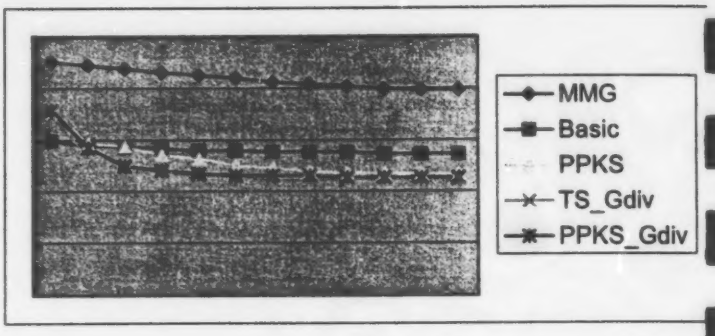
time	MMG Mean	Basic Mean	PPKS Mean	TS_Gdiv Mean	PPKS_Gdiv Mean
W10	BB =	339.9	*		
	0,00	0,00	0,00	0,00	0,00
	0,00	0,00	0,00	0,00	0,00
	0,00	0,00	0,00	0,00	0,00
	0,00	0,00	0,00	0,00	0,00
	0,00	0,00	0,00	0,00	0,00
	0,00	0,00	0,00	0,00	0,00
	0,00	0,00	0,00	0,00	0,00
	0,00	0,00	0,00	0,00	0,00
	0,00	0,00	0,00	0,00	0,00
	0,00	0,00	0,00	0,00	0,00
	0,00	0,00	0,00	0,00	0,00
W11	BB =	329,46	*		
	0,28	0,17	0,16	0,28	0,28
	0,28	0,17	0,13	0,28	0,28
	0,28	0,17	0,08	0,28	0,28
	0,28	0,17	0,05	0,28	0,28
	0,28	0,17	0,02	0,28	0,28
	0,28	0,17	0,02	0,28	0,28
	0,28	0,17	0,02	0,28	0,28
	0,28	0,17	0,02	0,28	0,28
	0,28	0,17	0,02	0,28	0,28
	0,28	0,17	0,02	0,28	0,28
	0,28	0,17	0,02	0,28	0,28
	0,28	0,17	0,02	0,28	0,28
W12	BB =	213,47			
	8,69	5,13	7,79	7,79	9,05
	8,13	3,66	5,62	5,24	6,14
	7,99	3,06	4,56	4,50	5,59
	7,33	2,99	4,13	4,09	4,69
	6,83	2,94	4,04	3,68	4,35
	6,52	2,89	3,98	3,27	4,16
	6,43	2,85	3,84	3,25	3,88
	6,29	2,85	3,78	3,23	3,78
	6,29	2,85	3,41	3,21	3,74
	6,25	2,85	3,24	3,21	3,60
	6,25	2,85	3,24	3,14	3,60
	6,25	2,85	3,24	3,05	3,50

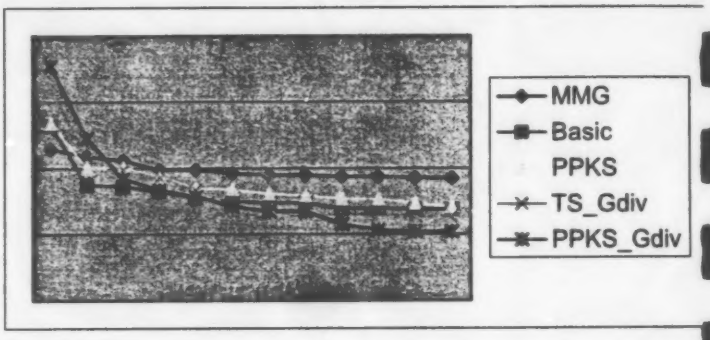


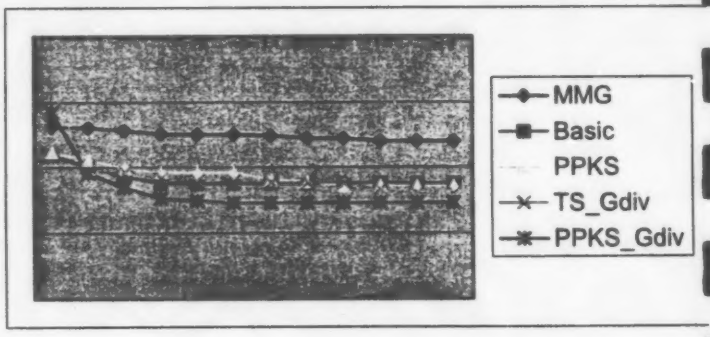




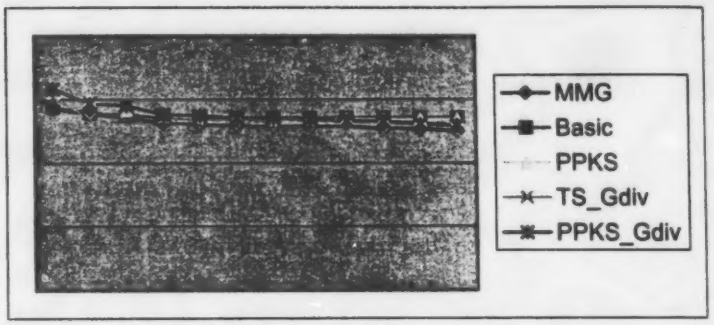
time	MMG Mean	Basic Mean	PPKS Mean	TS_Gdiv Mean	PPKS_Gdiv Mean
W13	BB =	241,2			
	9,01	5,98	7,08	6,92	7,14
	8,88	5,77	5,86	5,38	5,72
	8,76	5,74	5,75	4,77	5,00
	8,61	5,66	5,43	4,70	4,82
	8,50	5,59	5,26	4,63	4,69
	8,39	5,58	5,02	4,52	4,63
	8,23	5,53	4,90	4,45	4,62
	8,15	5,53	4,80	4,45	4,62
	8,06	5,52	4,71	4,44	4,52
	7,95	5,44	4,71	4,44	4,52
	7,93	5,44	4,69	4,44	4,52
	7,93	5,44	4,69	4,43	4,52
W14	BB =	228,8			
	4,57	5,28	5,36	6,42	7,09
	4,35	3,49	4,00	4,27	4,90
	4,23	3,43	4,05	3,95	3,69
	3,94	3,28	3,71	3,70	3,34
	3,94	3,10	3,38	3,32	3,14
	3,84	2,97	3,36	2,85	2,83
	3,79	2,95	3,22	2,81	2,69
	3,79	2,95	3,17	2,62	2,69
	3,74	2,76	3,07	2,62	2,30
	3,74	2,76	3,07	2,58	2,11
	3,69	2,76	2,93	2,54	2,07
	3,69	2,73	2,82	2,54	2,07
W15	BB =	263,91			
	5,24	4,38	4,46	6,43	5,68
	5,20	4,01	4,22	3,94	3,83
	5,13	3,73	3,99	3,87	3,37
	5,03	3,51	3,85	3,68	3,04
	5,00	3,51	3,85	3,58	2,97
	5,00	3,51	3,84	3,58	2,91
	4,96	3,51	3,62	3,53	2,91
	4,88	3,50	3,55	3,53	2,91
	4,88	3,50	3,44	3,53	2,90
	4,80	3,50	3,44	3,31	2,90
	4,80	3,50	3,43	3,19	2,90
	4,80	3,50	3,43	3,19	2,90

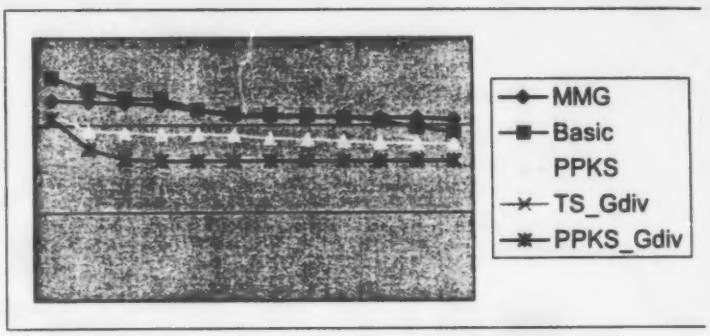


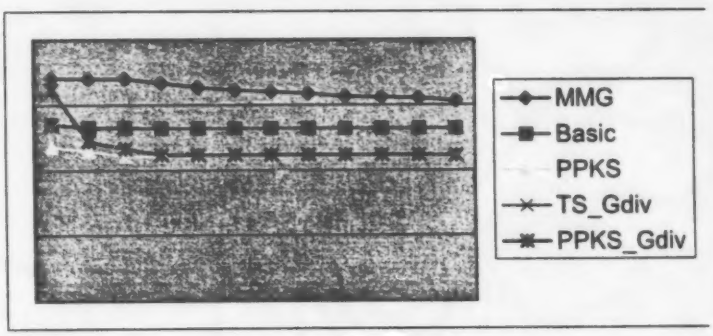




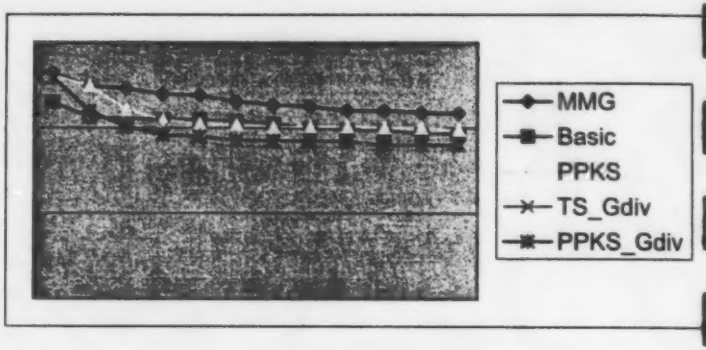
time	MMG Mean	Basic Mean	PPKS Mean	TS_Gdiv Mean	PPKS_Gdiv Mean
W16	BB = 236,37				
	14,30	14,38	15,53	15,19	15,84
	13,85	13,93	14,96	13,47	14,66
	13,59	13,75	14,11	13,17	14,48
	13,10	13,59	13,66	13,02	13,79
	13,01	13,54	13,58	12,79	13,71
	12,89	13,52	13,56	12,69	13,55
	12,85	13,52	13,46	12,61	13,53
	12,85	13,52	13,44	12,46	13,45
	12,85	13,52	13,43	12,41	13,31
	12,70	13,51	13,43	12,32	13,27
	12,70	13,51	13,43	12,31	13,05
	12,45	13,51	13,43	12,31	12,96
W17	BB = 198,53				
	11,44	12,72	10,60	10,76	10,41
	11,34	12,04	9,72	8,51	8,63
	11,25	11,52	9,57	8,20	8,07
	11,16	11,51	9,49	7,97	7,97
	10,86	10,92	9,48	7,97	7,96
	10,53	10,68	9,45	7,97	7,96
	10,47	10,60	9,24	7,97	7,96
	10,47	10,57	9,17	7,97	7,96
	10,47	10,42	9,09	7,96	7,96
	10,41	10,25	8,98	7,96	7,96
	10,41	9,88	8,92	7,96	7,96
	10,32	9,62	8,92	7,96	7,96
W18	BB = 247,39				
	17,03	13,52	11,77	13,20	16,17
	17,00	13,26	11,46	12,07	12,20
	17,00	13,23	11,07	11,83	11,65
	16,65	13,22	11,06	11,46	11,14
	16,33	13,21	11,06	11,32	11,14
	16,12	13,21	11,05	11,14	11,14
	15,98	13,21	11,05	11,14	11,14
	15,88	13,21	11,05	11,14	11,14
	15,64	13,21	11,05	11,14	11,13
	15,56	13,21	11,05	11,14	11,13
	15,52	13,21	11,05	11,14	11,13
	15,31	13,21	11,04	11,13	11,13

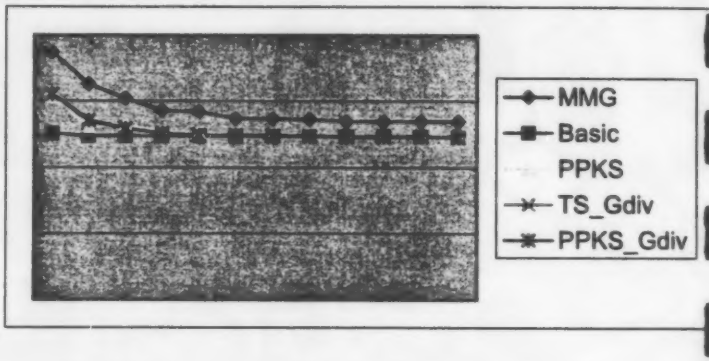


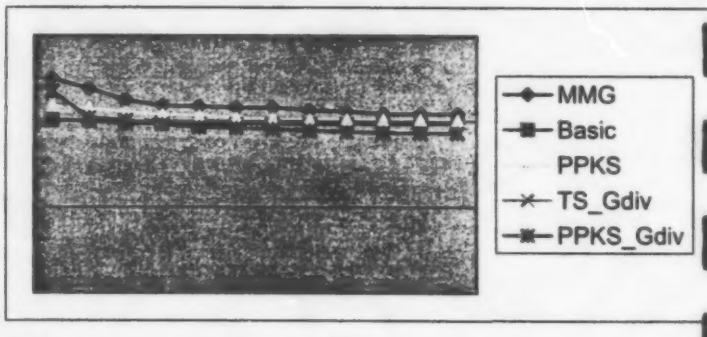




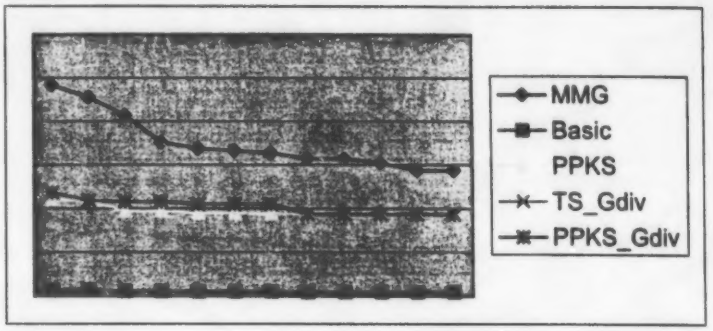
time	MMG Mean	Basic Mean	PPKS Mean	TS_Gdiv Mean	PPKS_Gdiv Mean
W19	BB = 205,18				
	13,20	11,52	13,13	12,86	12,99
	12,46	10,64	12,49	10,47	11,01
	12,33	10,56	11,05	10,35	10,10
	11,97	10,51	10,49	9,95	9,57
	11,92	10,51	10,27	9,24	9,52
	11,57	10,39	10,22	8,87	9,28
	11,36	10,28	10,05	8,84	9,25
	11,26	9,97	10,05	8,82	9,25
	11,04	9,91	10,05	8,74	9,19
	11,04	9,88	10,04	8,74	9,01
	10,89	9,68	10,00	8,69	9,01
	10,85	9,68	9,91	8,63	9,01
W20	BB = 256,59				
	7,48	5,02	6,15	6,46	6,21
	6,52	4,95	5,24	5,39	5,42
	6,09	4,95	5,29	5,09	5,18
	5,71	4,95	5,04	5,00	5,03
	5,67	4,95	5,04	4,98	4,98
	5,47	4,95	4,97	4,98	4,97
	5,47	4,95	4,97	4,98	4,95
	5,45	4,95	4,96	4,98	4,95
	5,38	4,95	4,96	4,97	4,92
	5,38	4,95	4,95	4,97	4,92
	5,38	4,95	4,95	4,97	4,92
	5,38	4,95	4,95	4,97	4,90
W21	BB = 179,32				
	12,50	10,09	10,90	11,42	11,75
	11,91	10,00	10,73	9,89	10,22
	11,23	9,89	10,54	9,70	10,04
	10,92	9,88	10,42	9,52	9,80
	10,86	9,86	10,33	9,46	9,64
	10,82	9,85	10,21	9,45	9,62
	10,80	9,85	10,15	9,45	9,53
	10,63	9,85	10,15	9,39	9,48
	10,59	9,85	10,14	9,31	9,39
	10,46	9,85	10,11	9,29	9,36
	10,46	9,85	10,11	9,27	9,33
	10,46	9,85	10,09	9,14	9,32

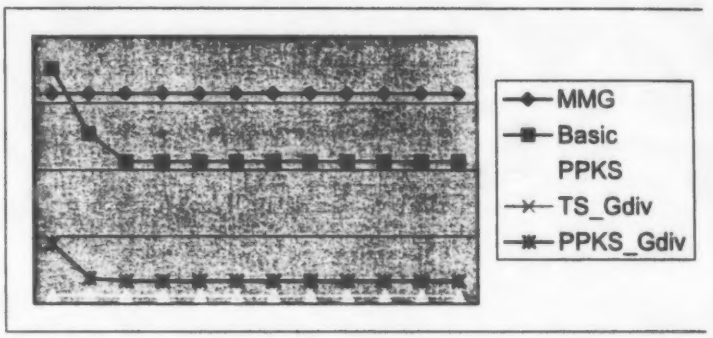




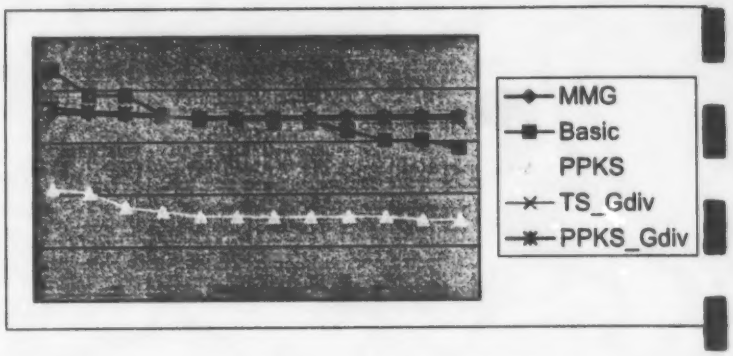


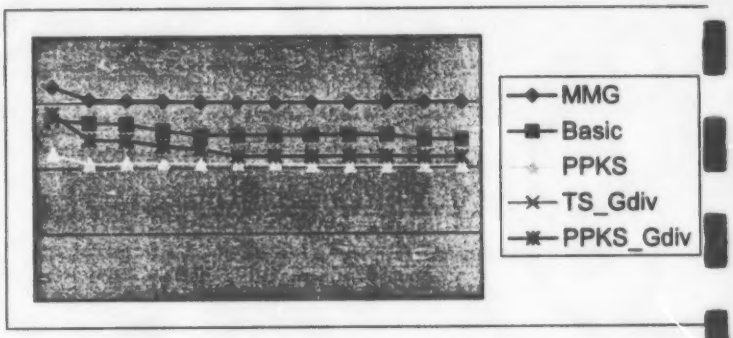
time	MMG Mean	Basic Mean	PPKS Mean	TS_Gdiv Mean	PPKS_Gdiv Mean
W22	BB =	278,14	*		
	4,83	0,12	2,22	1,82	2,37
	4,56	0,12	2,17	1,51	2,19
	4,15	0,11	1,95	1,51	2,16
	3,52	0,10	1,95	1,33	2,16
	3,38	0,09	1,89	1,32	2,11
	3,33	0,09	1,89	1,32	2,11
	3,27	0,09	1,89	1,14	2,10
	3,14	0,08	1,89	1,13	1,87
	3,14	0,08	1,89	1,13	1,87
	3,04	0,08	1,89	1,13	1,87
	2,88	0,08	1,88	1,13	1,87
	2,88	0,08	1,88	1,13	1,87
W23	BB =	196,92	*		
	1,57	1,76	0,06	0,52	0,44
	1,57	1,27	0,06	0,18	0,19
	1,57	1,06	0,06	0,16	0,16
	1,57	1,06	0,06	0,16	0,16
	1,57	1,06	0,06	0,16	0,16
	1,57	1,06	0,06	0,16	0,16
	1,57	1,06	0,06	0,16	0,16
	,57	1,06	0,05	0,16	0,16
	1,57	1,06	0,05	0,16	0,16
	1,57	1,06	0,05	0,16	0,16
	1,57	1,06	0,05	0,16	0,16
	1,57	1,06	0,05	0,16	0,16

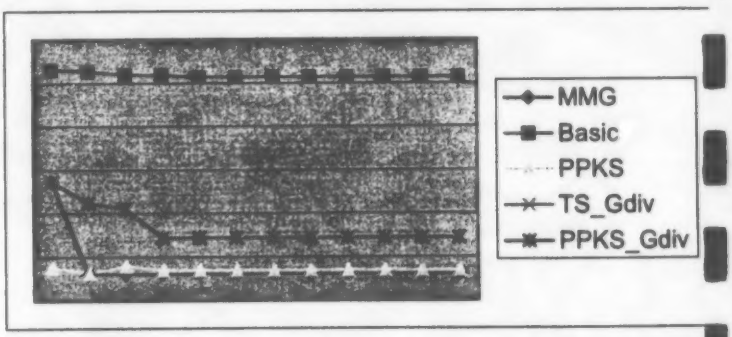




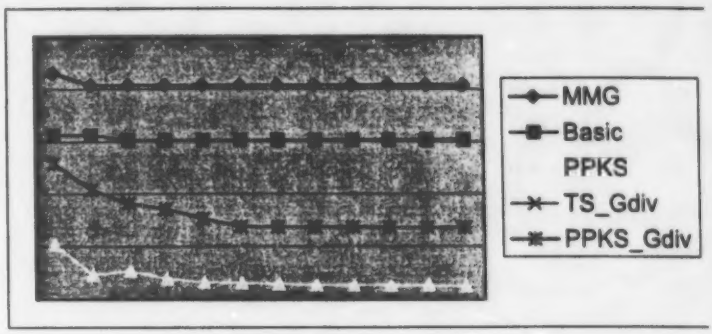
time	MMG Mean	Basic Mean	PPKS Mean	TS_Gdiv Mean	PPKS_Gdiv Mean
S1	BB =	130,27			
	1,79	2,19	1,06	1,91	1,84
	1,77	1,96	1,02	1,78	1,80
	1,76	1,95	0,88	1,77	1,78
	1,75	1,77	0,84	1,77	1,75
	1,74	1,71	0,79	1,74	1,73
	1,74	1,71	0,79	1,70	1,73
	1,74	1,69	0,79	1,69	1,72
	1,74	1,69	0,79	1,69	1,72
	1,74	1,60	0,79	1,67	1,72
	1,74	1,51	0,79	1,67	1,72
	1,74	1,51	0,75	1,67	1,72
	1,74	1,44	0,75	1,67	1,72
S2	BB =	133,63			
	6,49	5,49	4,43	5,29	5,68
	6,10	5,38	4,14	4,81	4,87
	6,06	5,36	4,16	4,45	4,86
	6,05	5,14	4,15	4,36	4,67
	6,00	5,01	4,14	4,36	4,58
	6,00	5,01	4,13	4,34	4,35
	5,99	5,01	4,13	4,34	4,34
	5,99	5,01	4,06	4,32	4,34
	5,99	5,00	4,05	4,32	4,34
	5,99	5,00	4,05	4,31	4,34
	5,99	4,86	4,05	4,31	4,34
	5,99	4,86	4,05	4,30	4,33
S3	BB =	139,43	*		
	1,36	2,65	0,38	1,08	1,36
	0,30	2,63	0,31	1,01	1,12
	0,30	2,59	0,40	1,01	1,05
	0,30	2,59	0,35	1,01	0,72
	0,30	2,57	0,35	1,01	0,72
	0,30	2,57	0,35	0,79	0,72
	0,30	2,57	0,35	0,79	0,72
	0,30	2,57	0,35	0,79	0,72
	0,30	2,57	0,35	0,79	0,72
	0,30	2,57	0,35	0,79	0,72
	0,30	2,57	0,35	0,79	0,72
	0,30	2,57	0,35	0,79	0,72

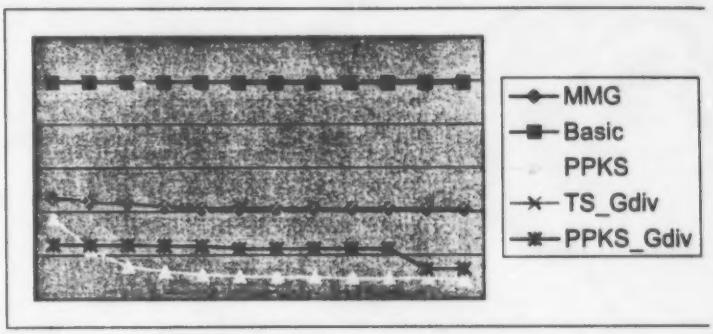


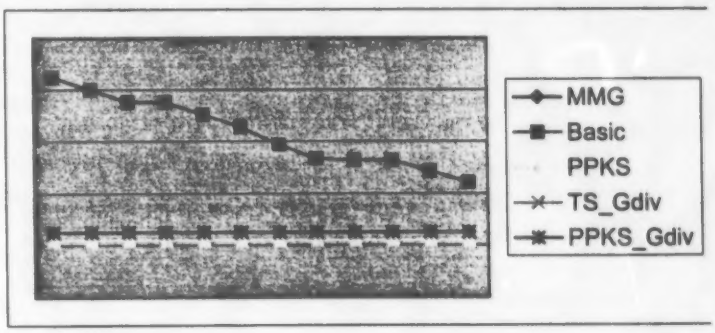




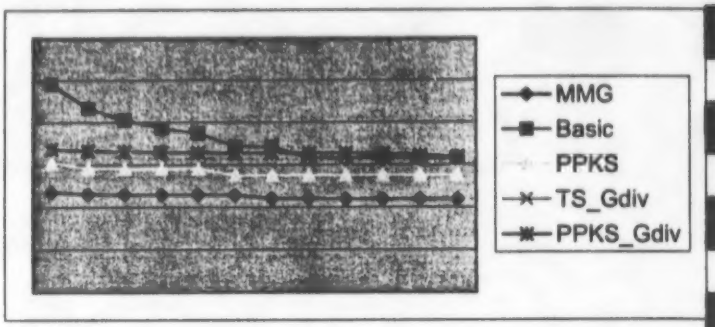
time	MMG Mean	Basic Mean	PPKS Mean	TS_Gdiv Mean	PPKS_Gdiv Mean
S4	BB =	122,39	*		
	2,15	1,56	0,52	1,08	1,29
	2,03	1,56	0,23	0,69	1,04
	2,03	1,51	0,27	0,60	0,91
	2,03	1,51	0,19	0,58	0,86
	2,03	1,51	0,16	0,52	0,77
	2,03	1,51	0,15	0,52	0,69
	2,03	1,51	0,15	0,52	0,68
	2,03	1,51	0,13	0,52	0,68
	2,03	1,51	0,13	0,52	0,68
	2,03	1,51	0,13	0,52	0,68
	2,03	1,51	0,12	0,52	0,68
S5	BB =	121,91	*		
	0,46	0,99	0,37	0,45	0,25
	0,44	0,99	0,22	0,45	0,25
	0,44	0,99	0,15	0,45	0,25
	0,42	0,99	0,13	0,44	0,24
	0,41	0,99	0,11	0,43	0,24
	0,41	0,99	0,10	0,43	0,23
	0,41	0,99	0,10	0,42	0,23
	0,41	0,99	0,10	0,42	0,23
	0,41	0,99	0,09	0,42	0,23
	0,41	0,99	0,09	0,42	0,23
	0,41	0,99	0,09	0,42	0,14
	0,41	0,99	0,09	0,42	0,14
S6	BB =	107,82			
	2,50	8,49	2,22	2,49	2,50
	2,49	8,01	2,21	2,49	2,49
	2,49	7,53	2,20	2,49	2,49
	2,49	7,53	2,20	2,49	2,49
	2,49	7,05	2,20	2,49	2,49
	2,49	6,56	2,20	2,49	2,49
	2,49	5,88	2,20	2,49	2,49
	2,49	5,34	2,20	2,49	2,49
	2,49	5,29	2,20	2,49	2,49
	2,49	5,29	2,20	2,49	2,49
	2,49	4,86	2,20	2,49	2,49
	2,49	4,44	2,20	2,49	2,49

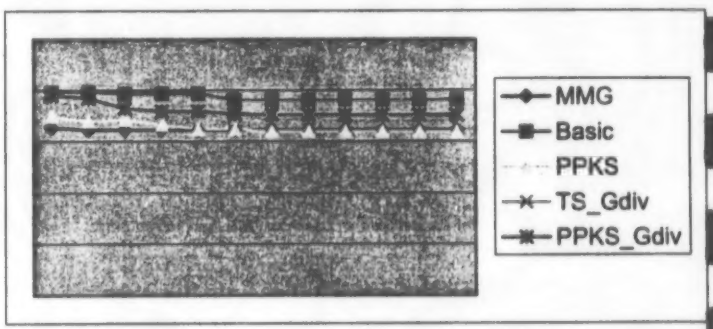


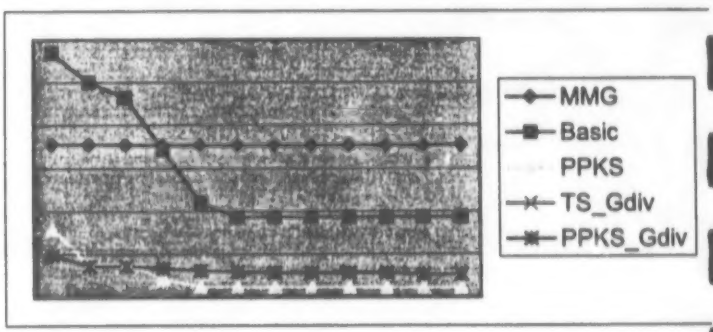




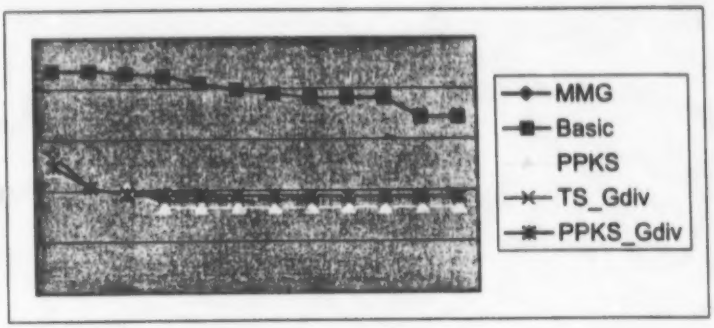
time	MMG Mean	Basic Mean	PPKS Mean	TS_Gdiv Mean	PPKS_Gdiv Mean
S7	BB =	139,62			
	2,36	4,91	3,05	3,40	3,36
	2,30	4,34	2,93	3,19	3,33
	2,29	4,06	2,91	3,16	3,30
	2,29	3,86	2,91	3,14	3,29
	2,28	3,74	2,90	3,14	3,29
	2,28	3,43	2,75	3,13	3,28
	2,19	3,42	2,75	3,13	3,28
	2,19	3,15	2,75	3,11	3,27
	2,19	3,15	2,75	3,11	3,27
	2,19	3,15	2,75	3,10	3,26
	2,19	3,15	2,75	3,10	3,22
	2,19	3,15	2,75	3,10	3,18
S8	BB =	143,41	*		
	6,53	7,93	7,00	7,60	7,81
	6,46	7,93	6,78	7,29	7,70
	6,46	7,88	6,86	7,27	7,30
	6,46	7,88	6,66	7,08	7,18
	6,46	7,88	6,48	7,05	7,18
	6,46	7,64	6,48	6,97	7,07
	6,46	7,64	6,40	6,87	7,04
	6,46	7,64	6,40	6,87	7,04
	6,46	7,64	6,40	6,78	7,03
	6,46	7,63	6,40	6,78	7,03
	6,46	7,63	6,40	6,78	7,03
	6,46	7,62	6,39	6,77	7,03
S9	BB =	144,65	*		
	1,77	2,85	0,79	0,59	0,48
	1,77	2,50	0,37	0,55	0,36
	1,77	2,32	0,39	0,55	0,36
	1,77	1,71	0,20	0,55	0,33
	1,77	1,10	0,12	0,53	0,31
	1,77	0,93	0,11	0,39	0,29
	1,77	0,93	0,11	0,31	0,28
	1,77	0,93	0,11	0,31	0,28
	1,77	0,93	0,11	0,31	0,28
	1,77	0,93	0,11	0,31	0,28
	1,77	0,93	0,11	0,31	0,26
	1,77	0,93	0,11	0,31	0,26

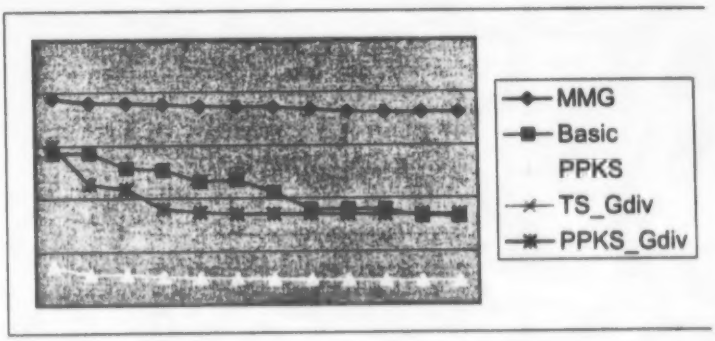


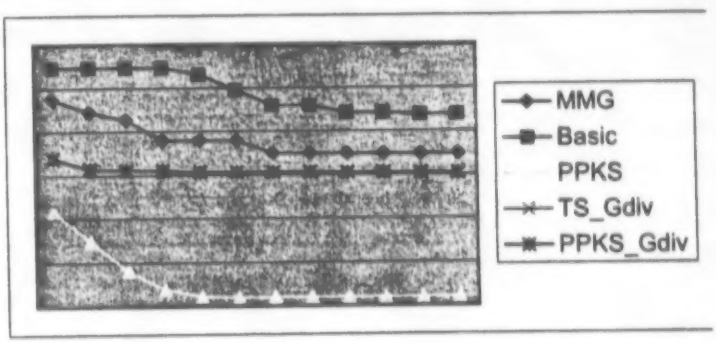




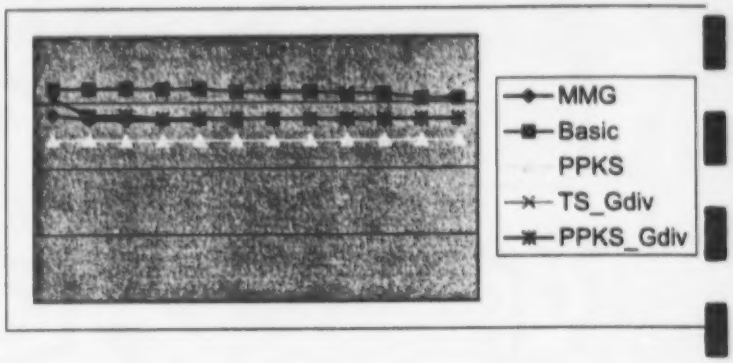
time	MMG Mean	Basic Mean	PPKS Mean	TS_Gdiv Mean	PPKS_Gdiv Mean
S10	BB =	139,1			
	2,76	4,36	2,42	2,75	2,49
	2,03	4,36	2,04	2,21	2,06
	2,00	4,30	2,04	1,94	1,93
	1,81	4,26	1,67	1,81	1,91
	1,80	4,11	1,67	1,81	1,90
	1,80	3,98	1,67	1,81	1,90
	1,80	3,90	1,67	1,79	1,90
	1,79	3,81	1,67	1,79	1,88
	1,79	3,81	1,66	1,79	1,87
	1,79	3,81	1,66	1,79	1,87
	1,79	3,43	1,66	1,78	1,87
	1,79	3,43	1,66	1,78	1,87
S11	BB =	134,42	*		
	1,94	1,43	0,36	1,07	1,50
	1,90	1,43	0,28	0,79	1,14
	1,90	1,28	0,28	0,70	1,09
	1,88	1,27	0,26	0,66	0,90
	1,87	1,16	0,25	0,66	0,87
	1,87	1,17	0,25	0,66	0,85
	1,86	1,05	0,24	0,65	0,85
	1,84	0,89	0,23	0,65	0,85
	1,81	0,89	0,23	0,65	0,85
	1,81	0,89	0,23	0,65	0,85
	1,81	0,82	0,22	0,65	0,85
	1,81	0,82	0,22	0,65	0,84
S12	BB =	146,65	*		
	2,37	2,73	1,08	2,23	1,69
	2,22	2,73	0,74	1,66	1,57
	2,13	2,73	0,40	1,48	1,56
	1,89	2,73	0,19	1,25	1,55
	1,89	2,65	0,11	1,25	1,53
	1,89	2,48	0,10	1,25	1,53
	1,75	2,30	0,10	1,21	1,53
	1,75	2,30	0,10	1,21	1,53
	1,74	2,21	0,10	1,21	1,52
	1,74	2,21	0,10	1,21	1,52
	1,74	2,19	0,10	1,21	1,52
	1,74	2,19	0,10	1,21	1,52

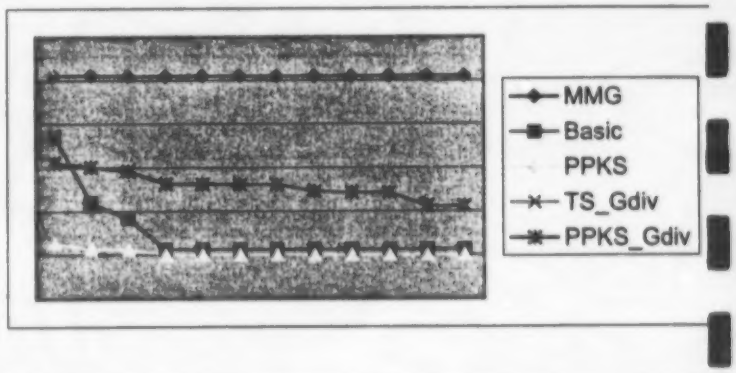


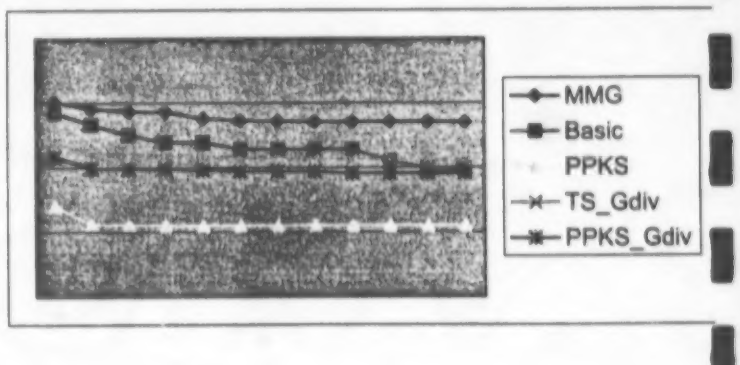




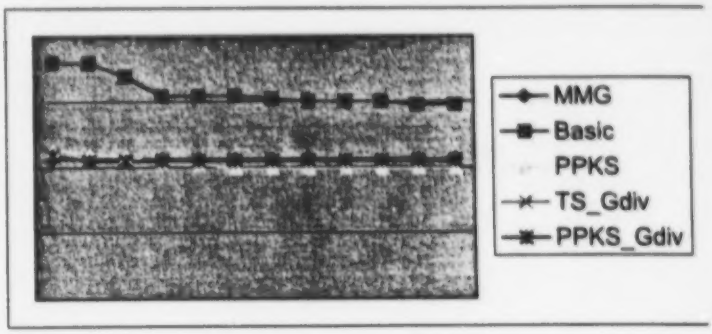
time	MMG Mean	Basic Mean	PPKS Mean	TS_Gdiv Mean	PPKS_Gdiv Mean
S13	BB =	153,11			
	14,03	16,03	12,17	14,84	15,29
	13,68	16,03	12,16	14,23	13,99
	13,66	16,03	12,15	14,05	13,98
	13,66	15,99	12,15	13,75	13,81
	13,66	16,03	12,14	13,67	13,81
	13,66	15,81	12,14	13,66	13,81
	13,66	15,81	12,14	13,65	13,76
	13,65	15,81	12,14	13,54	13,76
	13,63	15,66	12,13	13,53	13,76
	13,63	15,66	12,13	13,52	13,76
	13,63	15,26	12,13	13,52	13,76
	13,63	15,26	12,13	13,52	13,76
S14	BB =	135,53	*		
	2,06	1,36	0,13	1,41	1,05
	2,05	0,59	0,07	1,31	1,01
	2,05	0,41	0,06	1,14	0,96
	2,05	0,06	0,00	1,11	0,81
	2,05	0,06	0,00	1,11	0,81
	2,05	0,06	0,00	1,11	0,81
	2,05	0,06	0,00	1,10	0,81
	2,05	0,06	0,00	1,10	0,72
	2,05	0,06	0,00	1,10	0,71
	2,05	0,06	0,00	0,95	0,71
	2,05	0,06	0,00	0,95	0,56
	2,05	0,06	0,00	0,95	0,56
S15	BB =	117,3			
	3,02	2,85	1,40	2,39	2,17
	2,91	2,66	1,11	2,23	1,98
	2,86	2,50	1,10	2,22	1,97
	2,85	2,39	1,10	2,18	1,97
	2,75	2,39	1,10	2,18	1,95
	2,73	2,28	1,10	2,15	1,93
	2,72	2,28	1,10	2,15	1,93
	2,71	2,28	1,10	2,13	1,93
	2,71	2,28	1,10	2,12	1,90
	2,71	2,09	1,10	2,12	1,90
	2,71	1,98	1,10	2,12	1,90
	2,70	1,98	1,10	2,11	1,90

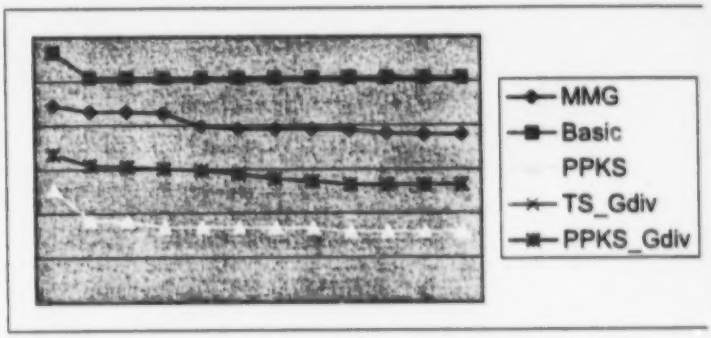


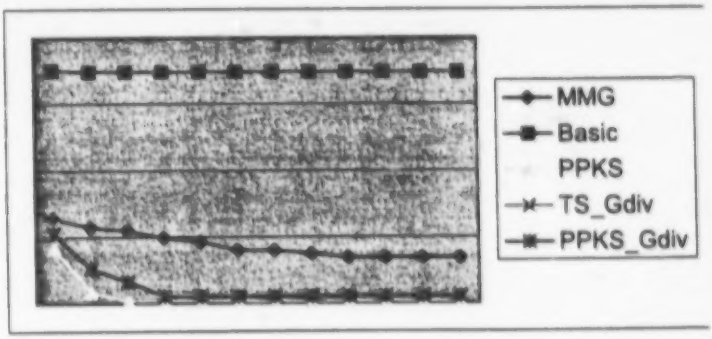




time	MMG Mean	Basic Mean	PPKS Mean	TS_Gdiv Mean	PPKS_Gdiv Mean
S16	BB =	110,05			
	4,25	7,19	4,52	5,09	4,37
	4,25	7,19	4,45	4,64	4,20
	4,25	6,78	4,29	4,64	4,19
	4,25	6,18	4,15	4,62	4,19
	4,25	6,18	4,09	4,62	4,19
	4,25	6,18	3,96	4,62	4,19
	4,25	6,09	3,95	4,61	4,19
	4,25	6,00	3,95	4,58	4,19
	4,25	6,00	3,95	4,44	4,19
	4,25	6,00	3,95	4,44	4,19
	4,25	5,91	3,95	4,32	4,19
	4,25	5,91	3,95	4,24	4,19
S17	BB =	150,36			
	2,22	2,83	1,27	1,72	1,67
	2,15	2,54	0,92	1,46	1,54
	2,15	2,54	0,91	1,44	1,53
	2,13	2,53	0,85	1,35	1,50
	1,98	2,53	0,85	1,27	1,49
	1,95	2,53	0,84	1,23	1,44
	1,95	2,53	0,84	1,23	1,39
	1,95	2,53	0,84	1,23	1,36
	1,94	2,53	0,80	1,23	1,32
	1,90	2,53	0,80	1,23	1,32
	1,89	2,53	0,80	1,23	1,32
	1,89	2,53	0,80	1,22	1,32
S18	BB =	132,21	*		
	2,55	7,02	1,66	2,32	2,10
	2,29	6,97	0,30	1,07	1,03
	2,19	6,97	0,05	0,63	0,64
	1,97	6,97	0,04	0,41	0,20
	1,85	6,97	0,04	0,27	0,20
	1,58	6,97	0,03	0,27	0,19
	1,58	6,97	0,03	0,25	0,19
	1,47	6,97	0,03	0,24	0,19
	1,35	6,97	0,03	0,24	0,19
	1,35	6,97	0,03	0,24	0,19
	1,35	6,97	0,03	0,24	0,19







time	MMG Mean	Basic Mean	PPKS Mean	TS_Gdiv Mean	PPKS_Gdiv Mean
S19	BB =	130,26	*		
	3,00	2,10	1,11	1,24	1,18
	2,59	2,08	0,97	0,61	0,34
	2,57	2,08	0,40	0,52	0,30
	2,53	2,08	0,22	0,42	0,30
	2,50	1,95	0,15	0,42	0,30
	2,47	1,95	0,05	0,42	0,29
	2,47	1,95	0,04	0,42	0,29
	2,47	1,95	0,04	0,42	0,29
	2,46	1,95	0,03	0,42	0,29
	2,46	1,95	0,02	0,42	0,29
	2,46	1,95	0,02	0,42	0,29
	2,46	1,92	0,01	0,42	0,21
S20	BB =	143,74			
	9,73	9,60	9,18	9,50	9,74
	9,44	9,60	8,74	8,90	9,15
	8,93	9,60	8,64	8,86	9,14
	8,72	9,44	8,54	8,86	9,08
	8,48	9,44	8,54	8,72	9,06
	8,48	9,36	8,54	8,72	9,06
	8,48	9,36	8,54	8,72	9,01
	8,46	9,20	8,54	8,67	8,89
	8,41	9,20	8,54	8,67	8,89
	8,40	9,20	8,54	8,65	8,86
	8,38	9,20	8,54	8,63	8,84
	8,38	9,20	8,54	8,54	8,84